

# The Minecart Project: A Wee Step Towards BI 2.0

---

Autor: Candón Arenas, Héctor  
Data de la defensa: 30/06/2014  
Director: Romero Moral, Oscar  
Departament del director: ESSI  
Codirector: Jovanovic, Petar  
Departament del codirector: ESSI  
Especialitat: Enginyeria del Software  
Centre: Facultat d'Informàtica de Barcelona (FIB)  
Universitat: Universitat Politècnica de Catalunya (UPC) – BarcelonaTech

---

# Resum

Tota empresa o negoci necessita, durant el transcurs del temps, prendre decisions estratègiques sobre afers que poden ser de gran rellevància per al seu futur. A mesura que el negoci creix, la importància d'aquestes decisions i la dificultat que comporta prendre-les augmenta.

El volum de les dades en les que ens podem basar per a dur a terme aquestes decisions pot ser enorme, i la naturalesa d'aquestes dades ens fa necessitar d'uns mecanismes especials per a poder tractar-les amb major eficiència i precisió.

Així, els magatzems de dades (*data warehouses*) s'encarreguen d'emmagatzemar aquestes dades i els processos d'extracció, transformació i càrrega de dades (ETL) s'encarreguen de recollir-les de les seves fonts originals i transformar-les per a donar coherència al sistema i assegurar-ne la consistència.

El disseny d'aquests elements resulta una tasca difícil i que requereix d'un alt grau d'expertesa per part de qui la du a terme. A mes a mes, consumeix una gran quantitat de temps i recursos.

Son aquests fets els que van motivar a un grup de recerca del departament d'Enginyeria del Software de la Universitat Politècnica de Catalunya a crear un sistema que permetés a qualsevol usuari no expert de crear, dissenyar, modificar, i desplegar els seus propis sistemes de data warehousing i els seus processos ETL, d'una forma senzilla, semi-automàtica i intuïtiva.

Quatre anys més tard alguns dels mòduls que conformen el sistema es troben en les últimes etapes del seu desenvolupament, arrel d'altres treballs de màster i de doctorat, però desenvolupen funcions separades i no existeix comunicació entre ells.

És aquí on neix la idea inicial del projecte: Un mòdul anomenat *Minecart* que tindria per missió actuar de repositori intern dels mòduls del projecte, actuar com a agent de control de flux i conciliar tota la feina en un sol software.

Amb el pas del temps, però, la idea es va fer encara més ambiciosa, de forma que *Minecart* no només ha resultat ser el repositori intern del software, sino un ecosistema sencer on s'ha transformat la solució inicial en una aplicació web, amb una interfície que permet a l'usuari interactuar amb una representació gràfica del domini de negoci o aplicació i dissenyar-hi els seus propis requisits d'informació de negoci d'una forma visual i intuïtiva, els quals representen el principal input per a dissenyar el seu magatzem de dades.

*El Projecte Minecart* representa, com el nom del treball indica, un petit pas sobre el *Business Intelligence 2.0*. Una eina que concilia un llarg treball de quatre anys i el transforma en un software visual, pràctic, intuïtiu i que permet a usuaris sense expertesa treballar i crear els seus propis ecosistemes *data warehousing* amb facilitat.

---

Toda empresa o negocio necesita, en algún momento, tomar decisiones estratégicas sobre diferentes asuntos que pueden ser de gran relevancia para su futuro. A medida que el negocio crece, la importancia de estas decisiones y la dificultad que comporta tomarlas aumenta.

El volumen de los datos en el que nos podemos basar para llevar a cabo estas decisiones puede ser enorme, y la naturaleza de estos nos hace necesitar de unos mecanismos especiales para poder tratarlas con mayor eficiencia y precisión.

Así, los almacenes de datos (o *data warehouses*) se encargan de almacenar estos datos, y los procesos de extracción, transformación y carga de los datos (ETL) se encargan de recogerlos de sus fuentes originales y transformarlos para dar coherencia al sistema y asegurar su consistencia.

El diseño de estos elementos resulta una tarea difícil y que requiere de un alto grado de experiencia por parte del que lo lleva a cabo. Además, consume una gran cantidad de tiempo y recursos.

Son estos hechos los que motivaron a un grupo de investigación del departamento de ESSI para crear un sistema que permitiese a cualquier usuario no experto crear, diseñar, modificar y desplegar sus propios sistemas de *data warehousing* y sus procesos ETL, de una forma sencilla, semiautomática e intuitiva.

Cuatro años más tarde algunos de los módulos que conforman el sistema se encuentran en las últimas etapas de su desarrollo, a raíz de otros trabajos de máster y doctorado, pero se encargan de funciones separadas y no existe comunicación entre ellos.

Es aquí donde nace la idea inicial del proyecto: Un módulo llamado Minecart que tendría por misión actuar de repositorio interno de los módulos del proyecto, actuar como agente de control de flujo y conciliar toda la faena en un solo software.

Con el paso del tiempo, la idea se hizo aún más ambiciosa, de forma que Minecart no solo ha resultado ser el repositorio interno del software, sino un ecosistema entero donde se ha transformado la solución inicial en una aplicación web, con una interfaz que permite al usuario interactuar con una representación gráfica del dominio de negocio o aplicación y diseñar ahí sus propios requisitos de información de negocio de una forma visual e intuitiva, los cuales representan el principal input para diseñar su almacén de datos.

El *Proyecto Minecart*, como el nombre del trabajo indica, representa un pequeño paso sobre el *Business Intelligence 2.0*. Una herramienta que concilia un largo trabajo de cuatro años y lo transforma en un software visual, práctico, intuitivo y que permite a los usuarios sin experiencia trabajar y crear sus propios ecosistemas data warehousing con facilidad.

---

Any business will need, among time, to take strategic decisions about different fields. These decisions can be of big relevance in a future. As long as the business grows, the relevance of these decisions and the amount of work needed to take them gets bigger.

The amount of data on which we can rely in order to take these decisions can be huge, and the nature of the data makes us need special mechanisms in order to treat them with more efficiency and precision.

This way, data warehouses are responsible to keep this data, and Extraction, Transformation and Load (ETL) processes are made in order to get the data from its original data sources and transform them to give coherence and consistency to the system.

The design of these elements is a very difficult task, which requires a high level of expertise. Furthermore, is a very time and resources consuming task.

This scenario motivated a research group of the Software Engineering department of the Polytechnic University of Barcelona (UPC) to create a system that could let any non-expert user to create, modify and deploy its own data warehouse systems and ETL processes in a simple, intuitive and semi-automatic way.

Four years later some of the modules that compose the system are in the last stages of its development. They were developed in other Master and PhD thesis, so they act independently and there is no communication between them.

At this point the idea of this project appears: A new module named Minecart that could act as an internal repository of the whole system, and as a flux control agent, conciliating all the work done in a single software tool.

Time passed and the idea became more ambitious, so Minecart grew as a whole ecosystem, transforming the initial solution in a web application with an interface that lets the user interact with a graphical representation of the business domain, and design its own business requirements in a visual and intuitive way. These requirements are the main input of the data warehousing design.

The Minecart Project represents, as the name of the thesis points, a wee step over Business Intelligence 2.0. A tool that conciliates a long work of four years and transforms it into a visual, practical and intuitive software that will let non-experienced users to design and create its own data warehouse ecosystems easily.

# Índex

1. Introducció .....	10
2. Estat de l'Art .....	11
2.1 Conceptes .....	11
2.1.1 Presa de Decisions .....	11
2.1.2 Data Warehouses .....	12
2.1.3 Processos ETL .....	14
2.2 Treball Relacionat: Projecte GEM .....	15
2.2.1 AMDO .....	15
2.2.2 GEM .....	16
2.2.3 ORE .....	16
2.2.4 COAL .....	17
2.3 Stakeholders .....	18
2.3.1 Usuaris Principals .....	18
2.3.2 Usuaris Afectats .....	18
3. Definició de l'Abast .....	19
3.1 Objectius i Context .....	19
3.2 Abast .....	21
3.3 Obstacles .....	22
3.4 Metodologia .....	23
4. Planificació Temporal .....	25
4.1 Calendari .....	25
4.2 Descomposició de Tasques .....	25
4.2.1 Fases del projecte .....	25
4.2.2 Tasques Principals del projecte .....	26
4.2.2.1 Tasques de la Fase 0 .....	26
4.2.2.2 Tasques de la Fase 1 .....	26
4.2.2.3 Tasques de la Fase 2 .....	27
4.2.2.4 Tasques de la Fase 3 .....	28
4.2.2.5 Estimacions Temporals .....	28
4.2.2.6 Consideracions .....	28
4.3 Diagrama de Gantt inicial .....	29
4.4 Control de Desviacions .....	30

5. Pressupost i Sostenibilitat .....	31
5.1 Identificació i estimació de costos .....	31
5.1.1 Recursos Humans .....	31
5.1.2 Hardware.....	31
5.1.3 Software .....	32
5.1.4 Despeses Generals .....	32
5.1.5 Pressupost Total.....	33
5.1.6 Viabilitat Econòmica.....	33
5.2 Control de Pressupost .....	34
5.3 Impacte.....	34
5.3.1 Impacte Social .....	34
5.3.2 Impacte Mediambiental .....	35
6. Inepció .....	36
6.1. Informes .....	36
6.2 Elicitació de Requisits.....	39
6.2.1 Vocabulari .....	40
6.2.2 Requisits .....	40
6.2.2.1 Interfície Principal del Sistema.....	40
6.2.2.2 MineDigger.....	50
6.2.2.3 Visor de resultats.....	57
6.2.2.4 Tasques Obertes.....	57
7. Implementació .....	59
7.1 Sprint 1 .....	59
7.2 Sprint 2 .....	60
7.3 Sprint 3 .....	62
7.4 Sprint 4 .....	63
7.5 Sprint 5 .....	66
7.5 Sprint 6 .....	67
7.7 Sprint 7 .....	70
7.8 Sprint 8 .....	74
7.9 Sprint 9 .....	75
7.10 Sprint 10 .....	79
7.11 Sprint 11 .....	83
7.12 Sprint 12 .....	85

7.13 Product Backlog Final .....	88
7.14 Gràfiques .....	91
8. Planificació Final .....	92
8.1 Desviacions .....	92
8.2 Diagrama de Gantt Final .....	93
8.3 Pressupost Final .....	94
9. MineDigger i Minecart <i>in a Nutshell</i> .....	95
9.1 Arquitectura .....	95
9.2 Tecnologies .....	98
9.3 Mapa Navegacional Final .....	99
9.4 Exemple de Flux de Treball .....	100
10. Testing .....	106
10.1 Dia de Test .....	106
11. Treball Futur .....	110
12. Conclusions .....	111
13. Glossari d'Abreviatures .....	112
14. Referències .....	113

# Índex de Figures

Figura 1: Esquemàtic d'un data warehouse .....	13
Figura 2: Exemple d'un esquema per a un data warehouse.....	13
Figura 3: Exemple d'un procés ETL.....	14
Figura 4: Esquema del procés AMDO .....	15
Figura 5: Arquitectura del sistema GEM .....	16
Figura 6: Exemple d'integració d'esquemes multidimensionals .....	17
Figura 7: Exemple d'unificació de fluxos ETL .....	17
Figura 8: Estat actual de la finestra de creació de requisits.....	20
Figura 9: Estat actual de la finestra de definició de focus d'anàlisi .....	20
Figura 10: Diagrama de Gantt inicial del projecte.....	29
Taula 1: Pressupost dels recursos humans .....	31
Taula 2: Pressupost dels recursos hardware.....	31
Taula 3: Pressupost dels recursos software .....	32
Taula 4: Pressupost de les despeses generals.....	32
Taula 5: Pressupost Total .....	33
Figura 11: Representació gràfica del Iron Triangle clàssic de la metodologia “en cascada” .....	33
Figura 12: Diagrama BPMN del requisit #001 .....	41
Figura 13: Diagrama BPMN del requisit #002 .....	42
Figura 14: Diagrama BPMN del requisit #003 .....	43
Figura 15: Diagrama BPMN del requisit #004 .....	44
Figura 16: Diagrama BPMN del requisit #004 .....	45
Figura 17: Diagrama BPMN del requisit #006 .....	46
Figura 18: Diagrama BPMN del requisit #007 .....	47
Figura 19: Diagrama BPMN del requisit #008 .....	48
Figura 20: Mapa navegacional inicial del sistema.....	49
Figura 21: Mockup de les funcionalitats del MineDigger.....	56
Figura 22: Exemple de parametrització errònia i visualització incorrecta del graf.....	61
Figura 23: Primera proposta d'estructura de la base de dades interna .....	63
Figura 24: Exemple de la visualització de la ontologia d'exemple TPC-H .....	64
Figura 25: Menú desplegable que apareix al fer clic dret sobre el node .....	66
Figura 26: Finestra contextual de configuració dels nodes .....	68
Figura 27: Esquema de la distribució client-servidor de la aplicació .....	71
Figura 28: Finestra modal de configuració dels nodes.....	71



Gràfica 1: Comparativa entre el temps real de treball i les estimacions .....	91
Gràfica 2: Diferència entre el temps real de treball i les estimacions .....	91
Figura 29: Diagrama de Gantt final del projecte .....	93
Taula 6: Pressupost Final .....	94
Figura 30: Arquitectura de la base de dades en relació amb els fitxers .....	95
Figura 31: Arquitectura de la base de dades en relació amb les col·leccions .....	96
Figura 32: Arquitectura completa del sistema .....	97
Figura 33: Mapa Navegacional final .....	99
Figura 34: Finestra principal del projecte .....	100
Figura 35: Finestra de creació de projecte .....	100
Figura 36: Finestra de gestió de projecte .....	101
Figura 37: Disseny d'un focus d'anàlisi amb el MineDigger .....	102
Figura 38: Disseny d'un <i>slicer</i> amb el MineDigger .....	102
Figura 39: Creació del requisit de negoci amb el MineDigger .....	103
Figura 40: Finestra de gestió de projecte .....	103
Figura 41: Finestra de càrrega de requisits al procés principal .....	104
Figura 42: Visor de resultats – Interpretacions multidimensionals .....	104
Figura 43: Visor de resultats – Processos ETL .....	105
Figura 44: Requisit a dissenyar a la segona part del dia de test .....	107

# 1. Introducció

Durant els mesos del febrer al juny s'ha desenvolupat i completat amb èxit el *Projecte Minecart*, un projecte emmarcat dins l'entorn del *Business Intelligence* i que assalta una de les problemàtiques associades amb la presa de decisions estratègiques dins els negocis.

Aquest document pretén ser un recull de tota la documentació i el conjunt d'*artifacts* relacionats amb el projecte, i com podem veure, conté tant documentació tècnica del projecte com documentació de gestió de projecte.

En primer lloc disposem d'una **contextualització** a la secció 2, on definirem els conceptes bàsics amb els que treballarem i donarem a conèixer l'actual estat de l'art en relació amb el nostre tema. En aquest apartat parlarem sobre la presa de decisions als negocis, els magatzems de dades o *data warehouses* i els processos d'extracció, transformació i càrrega de dades, i a continuació introduïrem tot el treball que s'ha portat a terme els últims anys en relació amb el projecte

Seguidament la secció 3 definim l'**abast** del projecte i els principals **objectius**, així com la **metodologia** que utilitzarem. Aquesta secció constitueix el punt de partida del nostre projecte. Resulta una secció bastant important, ja que considerem que la metodologia que hem utilitzat ha resultat sens dubte una peça clau per l'èxit del projecte, i la resta de la memòria es desenvolupa en funció d'aquesta.

A continuació, podem veure a la secció 4 la **planificació inicial** que vam dur a terme, amb una descomposició de tasques inicial que com veurem més tard patirà un seguit de canvis i modificacions a causa de la metodologia emprada.

La secció 5 la conformen els **pressuposts** del projecte i la seva justificació, i l'estudi de **sostenibilitat**.

Podem trobar tota la informació relacionada amb el desenvolupament i la gestió del projecte a les seccions 6 i 7, que documenten amb detall i exactitud les fases d'**inició** i **implementació** del projecte, així com una l'estat de la **planificació final** un cop acabat el projecte a la secció 8.

Per tal de poder mostrar els **resultats** i conciliar tot el que s'ha mostrat fins al moment, la secció 9 pretén mostrar els bàsics del projecte: l'arquitectura, les tecnologies emprades, un exemple del flux de treball...

A la secció 10 podem trobar com hem realitzat **activitats de test** durant el transcurs del projecte, així com un detall d'una sessió de test que vam realitzar amb un conjunt de persones interessades en el projecte.

Finalment comentar que aquest projecte ens ha obert portes a noves extensions i nous projectes per a seguir creixent, i podem trobar detall de quin són els plans de **treball futur** a la Secció 11, seguit de les **conclusions** i aportacions a nivell personal a la Secció 12.

## 2. Estat de l'Art

### 2.1 Conceptes

#### 2.1.1 Presa de Decisions

En qualsevol tipus d'organització o negoci és necessari, en un moment o altre, dur a terme un conjunt de tasques o prendre un seguit de decisions que tenen a veure amb l'emmagatzemament de dades. Estem parlant de tasques operacionals i decisions estratègiques.

Les tasques operacionals son aquelles tasques que es prenen en les activitats del dia a dia de l'organització. Son tasques amb efecte a curt termini, s'han de prendre ràpid i no solen comportar massa risc. Per posar un exemple, els inserts de cada tiquet de compra que genera un caixer de supermercat i que és composen d'un seguit de ítems.

Per altre banda, tenim les decisions estratègiques. Aquest tipus de decisions afecten a tota la companyia, i venen acompanyades de grans canvis i un risc bastant més elevat. Per a la presa d'aquestes decisions cal recollir i analitzar un gran conjunt de dades i extreure unes conclusions per aconseguir que el nostre negoci creixi. Per exemple, canviar el preu d'un dels nostres productes o triar el lloc més adient per a col·locar una nova sucursal.

Imaginem que som els propietaris d'una gran cadena de restaurants. Una de les tasques operacionals amb la que hauríem d'enfrontar-nos seria, per exemple, assignar cada dia les taules als cambrers, i una decisió estratègica seria dissenyar una promoció per als restaurants que tenim a Espanya basada en el comportament dels clients de tot l'any anterior.

Per descomptat, aquesta presa de decisions estratègiques no és pas arbitrària, ni tampoc es poden prendre per instint. Necessitem (i disposem) de suport de dades. Seguint l'exemple dels restaurants, tenim al nostre abast una quantitat ingent de dades que podríem analitzar, la majoria resultat de tasques operacionals: Totes les factures de tots els clients durant els últims 3 anys, les matèries primeres que s'han demanat des de cadascun dels restaurants, el nombre de clients que hi ha al restaurant depenent del dia i l'hora... Amb les eines adients totes aquestes dades poden ser tractades i emprades per a respondre qüestions que ens portaran a prendre les decisions correctes per a la nostra companyia.

En aquest escenari, però, tenim dos problemes importants en relació a les dades:

- No podem analitzar les dades amb una solució clàssica (esquemes SQL operacionals i transaccionals). El volum de les dades i la naturalesa de les preguntes que volem resoldre fan necessària l'existència d'un altre tipus de sistema d'emmagatzematge. Parlarem sobre això a la secció 2.1.2
- Aquest tipus de dades normalment prové de diferents fonts, amb una semàntica pròpia. Hem de conciliar dades molt heterogènies en una estructura única, i hem de ser capaços d'unificar aquestes fonts correctament. Seguint un altre cop amb l'exemple del restaurant, si volem calcular quants plats de "Sopa de Tomàquet" s'han preparat en tots els nostres restaurants, haurem de tenir en compte que a Espanya aquest concepte pot anomenar-se *Sopa de Tomate* o *Gazpacho*, i als Estats Units s'anomenarà *Tomato Soup*. Tot i que tenen noms diferents, son el mateix concepte.

Així, haurem de realitzar un procés de tractament de les dades des de les fonts fins al punt final d'emmagatzematge. En parlarem a la secció 2.1.3

## 2.1.2 Data Warehouses

Tal i com diu Bill Inmon a [1:1], un data warehouse o magatzem de dades és una col·lecció de dades orientades a l'usuari, integrades, amb variació temporal i no-volàtils, amb l'objectiu de donar suport al procés de presa de decisions.

D'aquí podem extreure que un data warehouse (o magatzem de dades) és un conjunt de dades que compleix les següents característiques:

- Son emprades per analitzar un **àrea temàtica concreta**.
- Provenen **de diferents fonts**. Recollim dades de fonts heterogènies i les fusionem de forma que conceptes iguals de diferents fonts queden agrupats en un únic concepte al magatzem de dades
- Son emmagatzemats com a **històrics**. A diferència dels sistemes transaccionals, els magatzems de dades guarden les variacions que sofreixen les dades a mesura que passa el temps, en lloc de sobre escriure-les.
- **Inalterable**. Quan quelcom queda emmagatzemat al data warehouse, hi romà sense possibilitat de canviar.

Son aquestes característiques les que fan que un data warehouse sigui perfecte per a l'anàlisi de dades. Tal i com es diu a [2:1], els ecosistemes data warehouse han estat estesesament reconeguts com a eines de suport per a presa de decisions estratègiques a entorns de negoci complexos.

El procés de disseny d'un data warehouse és llarg i complex. Tal i com podem veure a la figura 1, un magatzem de dades inclou diferents processos:

- El primer que es fa és la recollida de dades de diferents fonts. Normalment aquestes fonts son Sistemes Operacionals.
- Seguidament es du a terme el procés d'integració del qual hem parlat anteriorment.
- Finalment aquestes dades s'analitzen i s'utilitzen en processos de Data Mining, anàlisis OLAP, entre d'altres.

No és objectiu d'aquest document analitzar en profunditat el funcionament i els interns d'un data warehouse, de forma que només comentarem dos punts:

- El procés d'integració s'anomena procés ETL, que significa Extracció, Transformació i Càrrega de les dades (*Extraction, Transformation and Load*). Definirem aquest procés amb detall a la secció 2.1,3
- El tipus d'esquema que utilitza la base de dades és un esquema multidimensional. Podem veure'n un exemple a la Figura 2. Aquest esquema intenta representar els diferents conceptes utilitzant dimensions i fets, i el seu disseny depèn totalment dels *requisits de negoci* que tinguem. Les **dimensions** son les diferents perspectives des d'on volem fer anàlisi, i els **fets** son les mesures que volem analitzar.  
Com veurem a la secció 2.2 realitzar aquest disseny és una tasca difícil, sobretot quan el projecte acaba de començar.

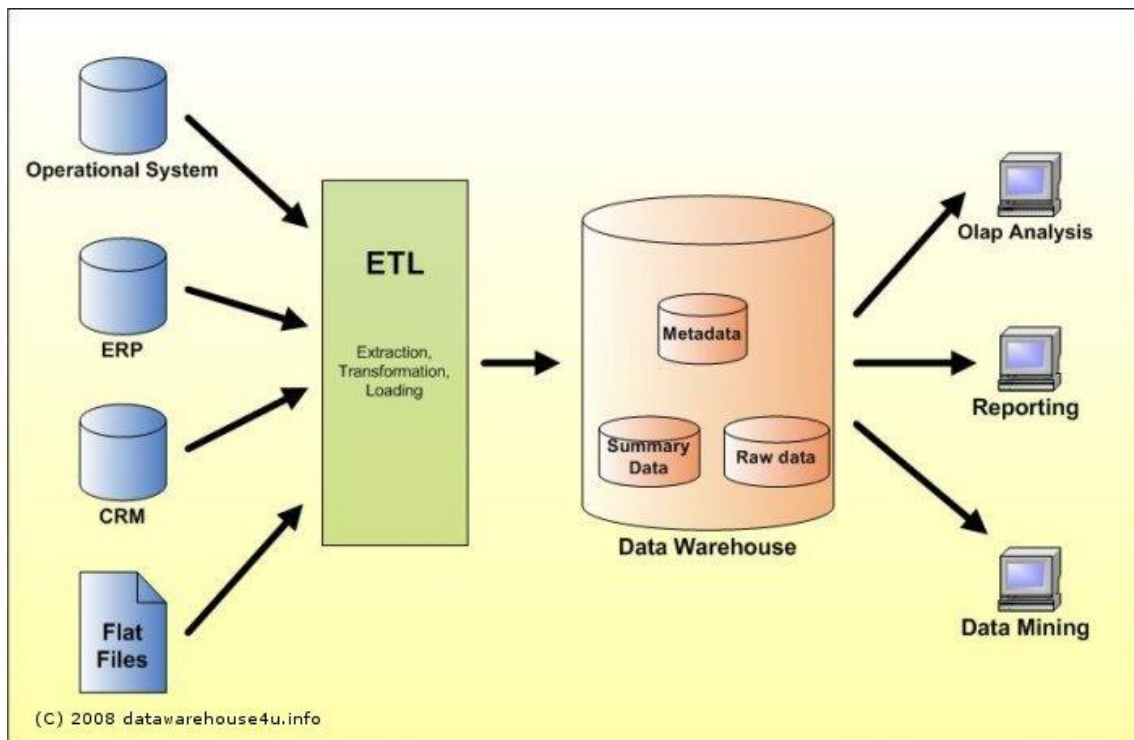


Figura 1: Esquemàtic d'un data warehouse [3]

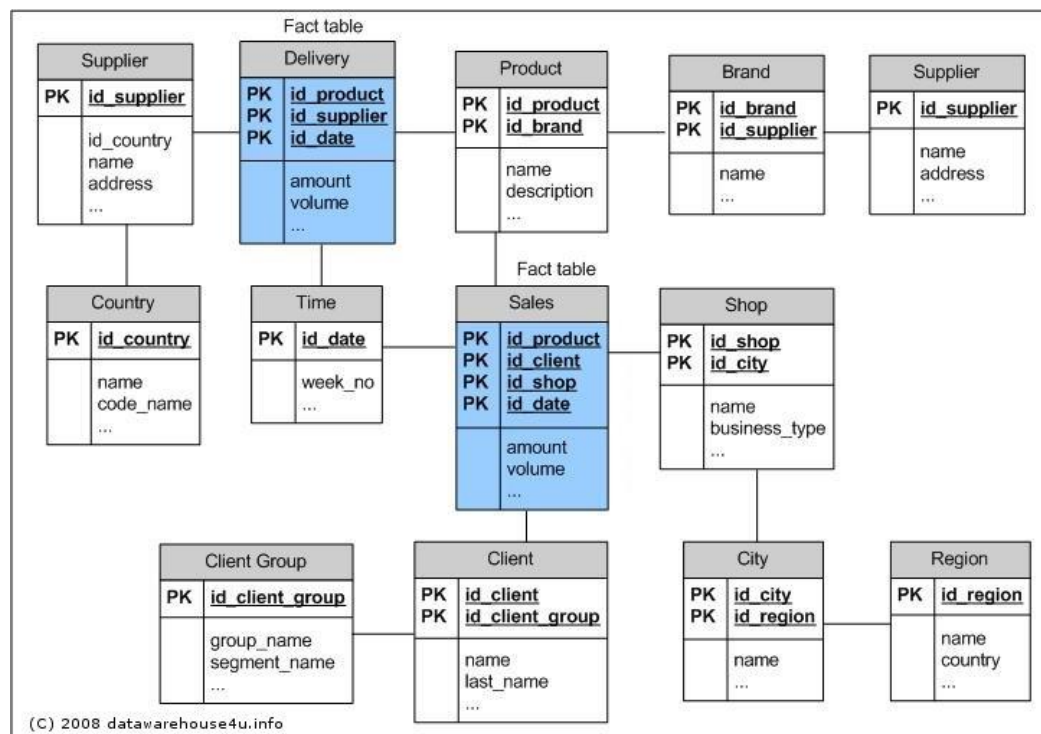


Figura 2: Exemple d'un esquema per a un data warehouse. Podem identificar com els diferents fets, dimensions i mesures formen un “esquema constel·lació”, un dels diferents tipus d'esquema Multi-Dimensional on algun dels fets comparteixen dimensions amb d'altres [4].

### 2.1.3 Processos ETL

Ja n'hem parlat diverses vegades en aquest document sobre el procés ETL i la fusió de dades de diferents fonts. A continuació explicarem breument en què consisteix aquest procés:

ETL son les sigles de *Extract*, *Transformation* i *Load*, els verbs que defineixen cadascuna de les tres fases de les que consta el procés.

- En primer lloc cal **extreure** les dades. Aquestes dades poden estar en diferents idiomes, notacions, abreviatures, formats... Una vegada acabada aquesta fase disposarem de una gran quantitat de dades.
- A continuació cal **transformar** les dades. Si per exemple hem recollit formularis de perfils, pot ser possible que el camp Gènere tingui dos valors: Home i Dona, però en un altre font ho representin amb les lletres H i D, o M i F (Masculí i Femení). Pot ser fins i tot possible que el nom del camp sigui Sexe i no Gènere. Un cop acabada la transformació haurem assolit **consistència**.
- L'últim pas és el procés de **càrrega** a la nostre base de dades (en el nostre cas un data warehouse).

Tot això queda il·lustrat a la figura 3, on podem veure un exemple de l'esquema d'un procés ETL. Podem identificar-ne les fonts a l'esquerra (en quadres verds), els passos de transformació (cercles blaus) i les taules del data warehouse on carregarem les dades transformades (quadres verds).

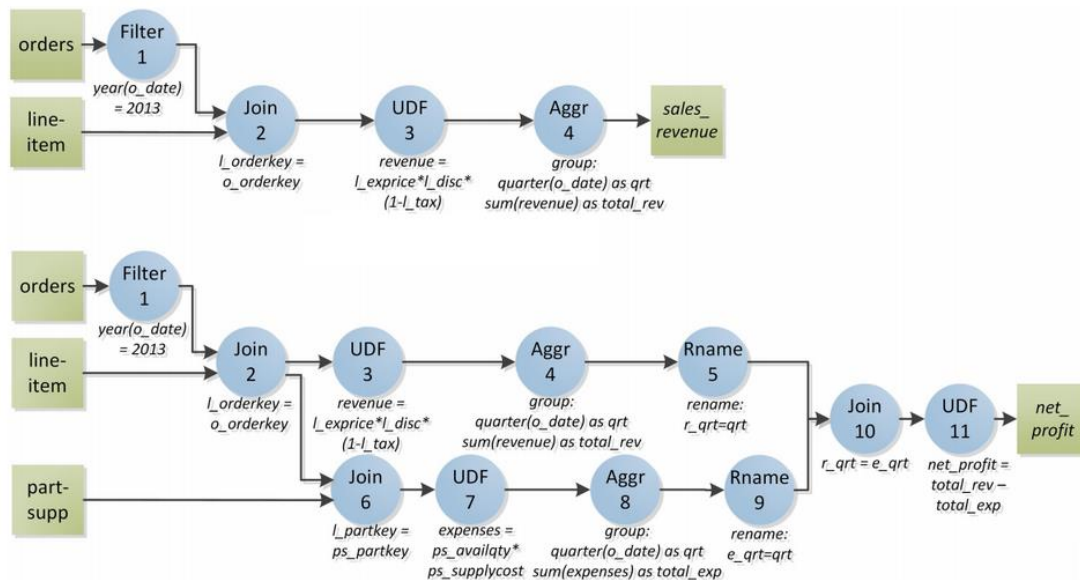


Figura 3: Exemple d'un procés ETL [5].

## 2.2 Treball Relacionat: Projecte GEM

Actualment no hi ha cap forma per a automatitzar completament la creació d'un data warehouse. Hi ha algunes aproximacions, però cap d'elles s'apropa suficient al objectiu [6, 7, 8, 9, 10] perquè requereixen d'interacció amb l'usuari i coneixement específic en molts dels passos del procés.

El projecte GEM intenta donar solució a aquest problema. Es tracta d'un projecte de caràcter de recerca que porta en desenvolupament des del 2010 a la Universitat Politècnica de Catalunya, a Barcelona. Ha estat desenvolupat per un conjunt de professors, estudiants de Màster i de Doctorat.

Aquest projecte es compon de diferents mòduls, cadascun d'ells desenvolupat com a un projecte independent, que s'encarreguen de dur a terme tasques concretes relacionades amb el disseny de data warehouses i ETL. Una vegada finalitzat, aquest conjunt de mòduls donaran forma a un software capaç de assistir a qualsevol persona no-experta, en el disseny i creació del seu propi data warehouse.

A continuació descriurem el conjunt de mòduls que componen a data d'avui el projecte GEM, explicant-ne l'objectiu i la funcionalitat.

### 2.2.1 AMDO

AMDO [11] (*Automating Multidimensional Design from Ontologies*) es una eina que dona suport al disseny d'esquemes multidimensionals a partir d'ontologies. Donada una ontologia, té per objectiu predir de forma totalment automàtica quins dels conceptes d'aquesta ontologia s'adaptarien millor com a fets, dimensions o mesures d'aquest esquema multidimensional. Utilitza tot un seguit d'algoritmes que acaben atorgant una puntuació a cada concepte. L'usuari rep aquests suggeriments i els modifica i adapta a la seva solució (si cal), passant doncs de requisits d'usuari d'alt nivell a requisits ja estructurats.

Hi ha hagut diferents sistemes que han intentat assolir aquest objectiu utilitzant esquemes UML com a punt d'entrada [6,7], però requereixen sempre un nivell d'interacció durant tot el procés. Utilitzant ontologies com a punt d'entrada, AMDO destaca per el seu grau d'automatització.

A la següent figura podem veure un esquema de tot el procés. Durant les diferents fases es van detectant els possibles rols de cada concepte. En primer lloc s'identifiquen les dimensions i seguidament els fets i les mesures.

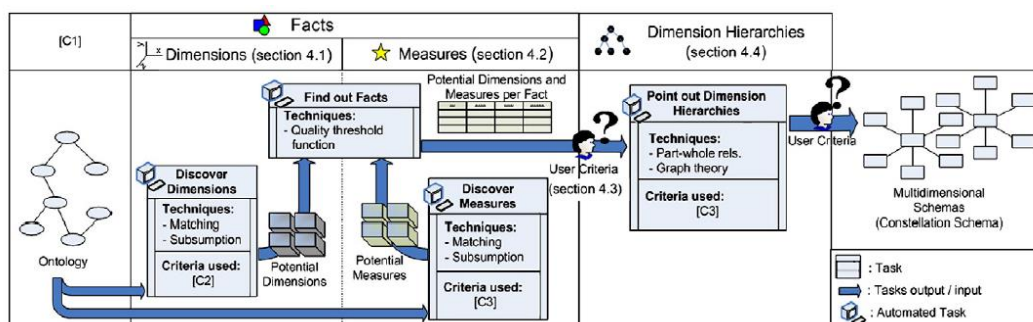


Figura 4: Esquema del procés AMDO [11]

### 2.2.2 GEM

GEM [12] (*Generating ETL and Multidimensional designs*) és el mòdul principal del projecte. Aquesta eina recull com a entrada un conjunt de requisits de negoci estructurats, una ontologia i un conjunt de *mappings* (dels conceptes de la ontologia als conceptes de les fonts d'entrada), i ens torna com a sortida dos elements:

- Un disseny conceptual de magatzem de dades que compleix els requisits proposats.
- Un conjunt de definicions dels diferents fluxos ETL necessaris per carregar el data warehouse.

En el següent esquema podem veure el flux de tot procés. Ara per ara aquests requisits de negoci es poden definir tant a la pròpia aplicació, com en documents separats seguint un format XML.

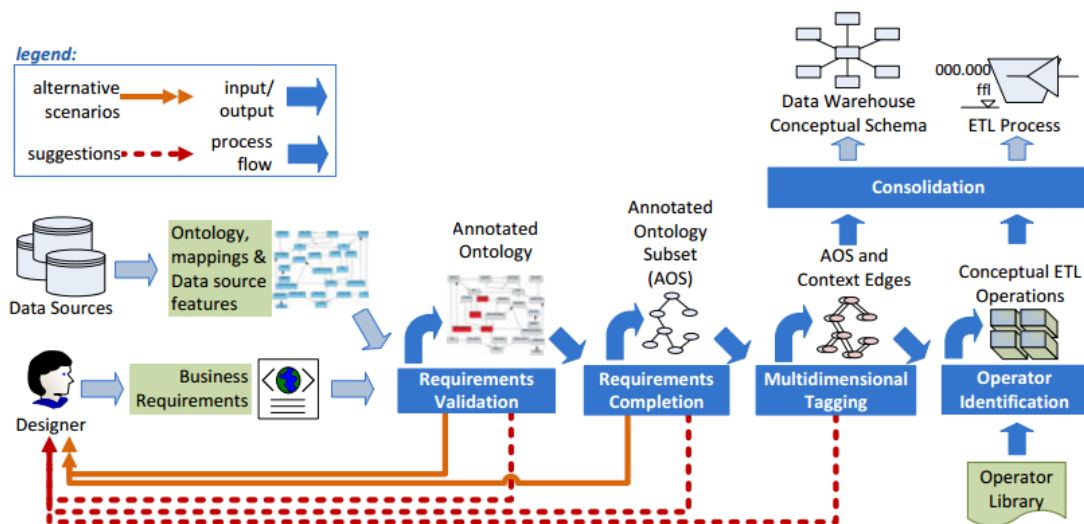


Figura 5: Arquitectura del sistema GEM [12]

### 2.2.3 ORE

El mòdul ORE (*Ontology-based data warehouse REquirement-driven evolution and Integration*) és descrit com un mètode semi-automàtic per a la integració i evolució de Requisits per a data warehouses basades en Ontologies. ORE assisteix la integració d'esquemes multidimensionals parcials (cadascun d'ells representant un requisit o una sèrie de requisits) en un esquema multi-dimensional unificat de forma semi-automàtica. [2:2]

Això significa que ORE pot rebre els outputs produïts pel GEM (esquemes multi-dimensionals parcials) i integrar-los en un de sol. Podem veure un esquema d'aquest procés a la figura 6: Tenim dos Esquemes MD que satisfan dos sets diferents de requisits de negocis (a) i després de passar pel mòdul ORE queden integrats (b).



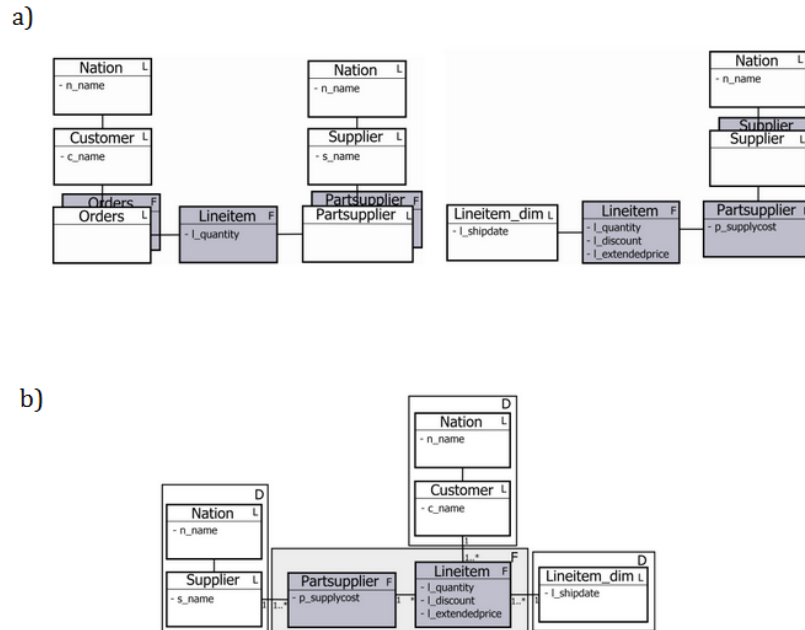


Figura 6: Exemple d'integració d'esquemes multidimensionals [2]

## 2.2.4 COAL

L'eina COAL (*ALgorithm for Consolidating ETL flows*) encara es troba en les primeres etapes del seu desenvolupament. COAL és, segons [5:1], un "algoritme per a consolidar fluxos ETL. COAL processa dissenys ETL parcials (satisfent cadascun un únic requisit de negoci) i els consolida en un disseny unificat que satisfà el conjunt sencer de requisits".

En aquest punt no podem dir massa sobre aquesta eina, ja que encara es troba en desenvolupament, però serà capaç de rebre els outputs del GEM i processar-los per a rebre un disseny de flux ETL unificat pel projecte. A la figura 7 podem veure com aquest procés arribarà a funcionar (Els passos on es produeix una unificació estan ressaltats).

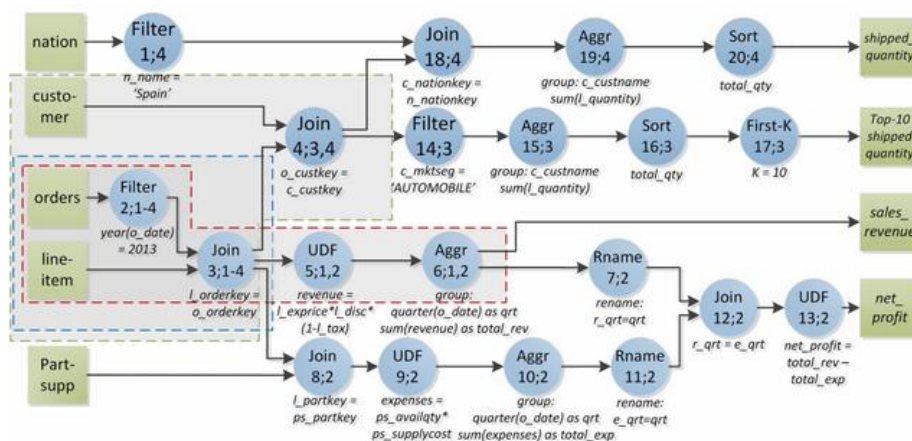


Figura 7: Exemple d'unificació de fluxos ETL [5]

## 2.3 Stakeholders

### 2.3.1 Usuaris Principals

Donat que aquest projecte es troba emmarcat dins l'àmbit de la recerca, no existeix un client directe pel qual estiguem desenvolupant el projecte. Tot i així, quan el projecte es trobi complet (no només el mòdul que estem tractant, sinó l'ecosistema complet), constituirà una eina que moltes companyies i negocis poden considerar útils, convertint-se així en clients potencials. Els avantatges del nostre projecte en relació a les companyies i negocis es troben definits a la Secció 5.3.1. Pot donar-se el cas que companyies que ja tenen el sistema establert i els seus propis data warehouses en funcionament no vulguin migrar o refer tot el seu sistema, però aquells negocis que acaben de crear-se, o es troben en auge poden arribar a estalviar molt de temps i recursos (tant humans com econòmics) si utilitzen el nostre software.

### 2.3.2 Usuaris Afectats

*L'Open Data* s'està tornant dia a dia més popular. Existeixen multitud de repositoris online amb dades obertes on cadascú pot consultar i emmagatzemar qualsevol cosa. Aquest món encara es troba en creixement, i la forma que ens ofereixen als usuaris per a interactuar-hi encara és massa bàsica i poc *user-friendly*. D'alguna manera el nostre projecte pot servir com a punt d'inspiració en aquest afer.

I no només gent de negocis pot trobar-se interessada en l'eina. Kristi Morton descriu al seu paper [13] la figura del *Data Enthusiast*, o entusiasta de les dades. Se'ns descriu un perfil de persona que no es dedica professionalment a l'anàlisi i el tractament de dades, però que gaudeix d'analitzar dades que recopila de diverses fonts (com els repositoris públics de dades). Per exemple, un grup de persones que s'autoanomenen *watchdogs* naveguen a través de fonts de dades públiques i intenten trobar evidències de casos de corrupció i destapar escàndols públics.

Aquesta gent no té ni les eines ni l'expertesa necessària per a extreure el màxim profit d'aquestes dades, però donat que la nostra eina serà usable per a usuaris no-experts, ells també en seran usuaris potencials.

## 3. Definició de l'Abast

### 3.1 Objectius i Context

Els Data Warehouses son, avui dia, una peça clau en el camp de l'anàlisi de dades, i una eina molt útil per a les companyies i empreses per tal d'extreure dades i informació rellevant.

El procés de disseny d'un data warehouse es llarg, complex, i requereix una gran quantitat de temps, diners i coneixement expert. Aquest és el motiu principal pel qual, a l'any 2010 neix el projecte GEM, amb l'objectiu de desenvolupar una eina per a assistir aquest procés.

Aquest projecte ha estat en desenvolupament durant els últims 3 anys a la Universitat Politècnica de Catalunya, de la mà d'alguns professors del departament d'Enginyeria del Software i estudiants de doctorat i màster.

L'objectiu principal del projecte GEM és donar capacitat als usuaris per a generar els seus propis dissenys i esquemes multidimensionals i processos ETL per a data warehouses. En altres paraules, una eina per assistir qualsevol usuari que vulgui construir-se el seu propi data warehouse, basant-se en les seves necessitats i requisits de negoci.

Disposant de requisits de negoci i fonts de dades com a entrades, i utilitzant ontologies com a conceptualització del domini, s'efectua un procés que té 4 passos (detallats a la secció 2.2).

- AMDO: Passem de requisits d'alt nivell definits per l'usuari a requisits estructurats.
- GEM: Passem de requisits estructurats al disseny conceptual del data warehouse i els processos ETL associats. Aquestes seran les entrades de l'ORE i el COAL.
- ORE: Construeix una proposta d'esquema multidimensional que satisfà un conjunt de requisits d'usuari.
- COAL: Proposa un disseny ETL que integra un conjunt de dissenys ETL d'entrada.

Durant els últims 3 anys el projecte ha anat creixent i alguns dels mòduls principals ja son funcionals. Tot i així el projecte encara es troba en desenvolupament, i noves necessitats i problemes han anat apareixent. Explicarem aquests problemes pas a pas.

Primerament, totes les peces del projecte s'han anat desenvolupant com a mòduls individuals, ja que formaven part de projectes de màster i tesis de doctorat, com a proves conceptuais dels treballs. Per suposat, el projecte final necessita de tots els mòduls treballant alhora i necessita que siguin capaces de comunicar-se.

D'aquesta manera, necessitem integrar els mòduls per a assolir interoperabilitat. Es a dir, hem de treballar amb **metadades** i **control de flux**.

En altres paraules, haurem de dissenyar i desenvolupar una base de dades interna per tal de que els mòduls es puguin comunicar entre si, així com formalitzar els fluxos d'informació entre els mòduls i la base de dades.

Aquesta base de dades tindrà també associada una sèrie de parsers que transformaran tots els objectes que s'utilitzen durant els processos en algun tipus de document que serà

interpretable per tots els mòduls del projecte. Desenvoluparem interfícies i APIs per la base de dades. Així, estem parlant d'un **repositori de dades** intern per al sistema.

El següent problema és aquest: Per tal de crear els dissenys per al data warehouse necessitem que els usuaris indiquin al sistema quins són els seus requisits de negoci (per exemple, una cadena de restaurants amb centenars de restaurants al voltant del món poden voler fer un agregat de les ventes des del punt de vista de la seva situació geogràfica).

Aquesta és la tasca més dura que demanem a l'usuari. Ara per ara disposem d'una simple interfície que és la responsable d'aquest afer (Figures 8 i 9). Ha estat desenvolupada en Java, utilitzant la llibreria gràfica Swing [14], i degut a això presenta un seguit de mancances i limitacions.

Per la naturalesa de la tasca, generar un requisit de negoci a partir d'aquesta interfície representa una tasca difícil per a l'usuari, ja que ha de navegar per llistes desplegable i caixes de text, de forma poc intuïtiva i sense cap representació gràfica, i a més gran és el domini conceptual, més difícil és de muntar aquest requisit.

Una forma molt més natural de fer aquesta tasca seria representant la ontologia del projecte en forma de graf (enlloc de llistes desplegable) i deixar que l'usuari pugui navegar-hi, definir els rols dels conceptes, veure les relacions d'aquells conceptes amb els que està treballant, definir agregats i funcions... I veure en conjunt quin aspecte té el requisit que està dissenyant.

Aquesta funcionalitat serà el nucli del projecte de final de grau, la tasca més important. Anomenem aquest nou mòdul **MineDigger**

Figura 8: Estat actual de la finestra de creació de requisits [Font: Propi]

Figura 9: Estat actual de la finestra de definició de focus d'anàlisi [Font: Propi]

Quan s'executa tot el procés es generen un seguit de fitxers de sortida. Aquests fitxers son grafs multidimensionals i processos ETL. Actualment es visualitzen també en una interfície feta amb Swing. Necessitarem un **visor de resultats** que mostri aquests fitxers a l'usuari i el permeti descarregar-los al seu ordinador.

Finalment cal agrupar tot aquest treball sota una **interfície** comú, i en una **arquitectura** adequada. D'aquesta forma, l'usuari serà capaç de crear els seus projectes, dissenyar els requisits de negoci que vol tractar, guardar i recuperar el seu treball... tot sota el mateix software.

Si aconseguim assolir aquestes tasques, doncs, disposarem d'un software totalment funcional.

## 3.2 Abast

Tal i com hem discutit a la secció anterior, volem implementar un seguit de mòduls nous per al projecte GEM. A continuació especificarem amb detall què volem aconseguir amb cada mòdul, i quins son els obstacles que ens podríem trobar durant el procés.

Primerament canviarem l'**arquitectura** del projecte sencer. Eliminarem per complet la interfície actual i proposem una nova solució: Convertir tot el projecte en una aplicació web. D'aquesta manera aconseguirem donar accés a la eina des de qualsevol dispositiu i en qualsevol moment.

D'aquesta manera, caldrà redissenyar el projecte per tal d'adaptar-lo a aquesta nova solució. A la banda del **servidor** mantindrem els mòduls funcionals del projecte (AMDO, GEM, ORE i COAL), així com el **Minecart**, i a la banda del client trobarem la **interfície gràfica**, el **Visor de Resultats** i el **MineDigger**.

Minecart serà un punt de contacte entre la banda del client i la del servidor, de forma que els elements que es generin a les interfícies passaran al servidor a través de Minecart, i els fitxers que s'hagin de mostrar als usuaris es recolliran del servidor també a través de Minecart.

Així doncs, per cada mòdul haurem de dur a terme les següents tasques.

Pel **repositori** haurem de:

- Triar el tipus de base de dades que utilitzarem i dissenyar-la base de dades.
- Implementar un conjunt de APIs per tal d'establir comunicació amb la banda del client i la del servidor.
- Implementar un conjunt de pàrsers per tal de poder manegar tots els diferents conjunts de dades amb les que treballen tots els mòduls del software.

Pel **MineDigger**, aquestes son les funcionalitats principals que volem aconseguir:

- L'usuari podrà carregar ontologies i mostrar-les a la pantalla. Aquestes ontologies son representades en forma de graf interactiu, de forma que l'usuari serà capaç d'interactuar-hi, moure'l, realitzar accions bàsiques (clic esquerre i dret, entre d'altres).
- L'usuari podrà indicar quins conceptes son del seu interès. Sobre aquests conceptes s'hi podrà definir el seu rol multidimensional (fets, dimensions o mesures).
- L'usuari podrà definir funcions sobre els conceptes (agregats, restriccions...) de forma dinàmica i gràfica.

- En qualsevol moment l'usuari podrà extreure i guardar un requisit de negoci a partir del seu disseny sobre el graf interactiu. També podrà carregar de nou sobre el graf requisits que ja hagi dissenyat.
- La finestra de treball tindrà tot un conjunt de funcionalitats que milloraran la experiència de l'usuari: personalització de la paleta de colors, diferents modes de visualització... així com altres eines típiques d'una aplicació web.

I finalment pel **Visor de Resultats** necessitem:

- Un visor capaç de carregar fitxers generats per l'eina i mostrar-los a l'usuari de forma gràfica (per exemple, mostrar els processos ETL d'una forma similar a com ho fa Kettle[15]).
- Que permeti a l'usuari descarregar els grafs multidimensionals com a sentències SQL i els processos ETL com a fitxers *Kettle*.

### 3.3 Obstacles

Donat que tenim un gran nombre d'objectius a assolir, hem realitzat un petit estudi intentant detectar els obstacles principals amb els que ens podem trobar durant el desenvolupament del projecte:

- Els mòduls de la banda del client molt probablement hauran de desenvolupar-se emprant eines per a aplicacions web com JavaScript, HTML i CSS com a llenguatges de programació. Donat que és el primer cop que desenvolupem una aplicació web i el primer cop que treballem amb aquests llenguatges, les estimacions de càrrega de treball poden ser errònies (donada la nostra falta de coneixement a l'àmbit). També pot donar-se el cas de que algunes de les funcionalitats que hem pensat en un principi siguin impossibles (o massa difícils) d'implementar, o les llibreries que utilitzem presentin limitacions.
- Esperem que la base de dades sigui estable i tingui uns temps de resposta acceptables, i per aquest motiu ens plantejem poder emprar tecnologies no relacionals (NoSQL). Aquest tipus de solucions són inestables, de forma que podem obtenir resultats inesperats.
- Donat que estem treballant amb gairebé tres anys de codi generat per multitud de persones, pot donar-se el cas de que no disposem del suficient coneixement per entendre parts d'aquest, que algunes funcionalitats no funcionin com esperem o que algunes parts del codi no siguin senzilles d'integrar amb d'altres.
- Tenim moltes tasques per a complir, i necessitem que quasi totes siguin totalment funcionals per tal de que el projecte funcioni. Una mala planificació o metodologia pot portar-nos a no acabar el projecte a temps.

### 3.4 Metodologia

La metodologia que hem triat per a desenvolupar el nostre projecte resulta clau per a l'èxit d'aquest. Donat que es tracta d'un projecte relacionat amb la recerca i que els requisits no es troben definits al començament, hem decidit utilitzar *Agile Software Development*[16]. A continuació definirem i justificarem aquesta metodologia.

Hem decidit dividir el projecte en quatre fases de la següent manera:

En primer lloc ens centrarem en dur a terme una detecció i definició de requisits. Durant el primer mes de projecte compaginarem aquesta fase amb GEP. El nostre objectiu principal és el de generar bona documentació, definint amb prou detall aquells requisits base amb els que començarem a treballar (elicitació), els mapes navegacions de la interfície, i diagrames de flux BPMN per tal de definir els diferents processos de forma intuïtiva. També generarem diferents *mockups* que mostraran l'aspecte visual d'alguns dels requisits. Aquestes tasques conformen les fases 0 i 1 (Secció 7)

Un cop finalitzada aquesta part, començarà la implementació. L'ús d'*Agile* en aquesta part del projecte serà la clau de l'èxit, ja que es tracta d'un projecte molt **susceptible als canvis**.

Com diu el *manifesto*[17] d'*Agile*, ens enfrontem amb un projecte que compleix les següents característiques:

- Baixa predictibilitat.
- Requisits que no estan suficientment clars al principi del projecte.
- Canvis continus a l'especificació de requisits.
- Estimacions imprecises.
- Temps elevat per passar de la idea al producte.
- Cal mostrar una visibilitat real de progrés.

Necessitem fer una implementació iterativa, amb un tracte constant amb el client (que redefinirà els requisits i les funcionalitats a partir de reunions setmanals i *demos* tècniques).

D'aquesta manera, *Agile* ens permet triar quines variables del projecte fixem i quines deixem obertes. En el nostre cas fixarem la data d'entrega, que és l'única variable que sabem del cert, i deixem el pressupost i els requisits com a variables. A mesura que avancin les iteracions el software prendrà forma i evolucionarà segons les necessitats del client fins a aconseguir un software funcional.

Seguirem els principis que dicta la metodologia *Agile*, i treballarem de la següent manera:

- A partir dels requisits inicials crearem el *product backlog* inicial, on documentarem tot el seguit de *user stories*. Les *user stories* son funcionalitats que ha de tenir el software i que deriven dels requisits o que apareixen arrel dels *sprints*.
- Aquestes *user stories* tenen definit un criteri d'acceptació, que s'acorda amb el client. Quan el criteri es compleix (el software pot dur a terme la funcionalitat definida) es

marca com a acceptada. Aquestes *user stories* poden ser modificades o cancel·lades amb prèvia justificació.

- Cada *user story* és acompanyada per una estimació temporal. Al final de cada *sprint* s'avalua el temps real que ha costat d'implementar, de forma que a mesura que el projecte avança les estimacions son més acurades i es pot advenir quina quantitat de feina es podrà fer a cada *sprint* (i planificar millor el projecte).
- El treball es divideix en *sprints*. Per cada *sprint* se celebra una reunió amb una demo tècnica (prototip), on es mostra la implementació d'un subconjunt de *user stories* del *product backlog*.
- A la demo tècnica de l'*sprint* es discuteix sobre els resultats, i cada *user story* tractada pot ser:
  - Acceptada
  - Modificada (un nou criteri d'acceptació pot ser necessari)
  - Rebutjada. Si es rebutja una *user story* pot donar lloc a un nou conjunt de *user stories* o a la modificació de les ja existents.

També ens centrarem en dur a terme un treball de test important. Com hem comentat cada setmana disposarem d'un prototip del programa que podrem testar i avaluar (els desenvolupadors i els clients). Quan s'apropi la data d'entrega durem a terme un dia de test amb un conjunt de participants que, directa o indirectament, han participat en el projecte GEM i coneixen el seu objectiu i el tipus d'usuari al que va destinat. A la secció 10 ho podem veure amb més detall.

Per últim, la duració dels nostres *sprint* serà d'una setmana. Cada *sprint* anirà acompanyat d'una reunió, de la qual en generarem un informe. Aquest informe pretén instaurar una reflexió després de cada *sprint* per veure què ha funcionat i no ha funcionat i quines millores es poden introduir per assolir l'excel·lència tècnica i un millor funcionament de l'equip de treball. Cal tenir en compte que es possible que en alguns punts calgui treballar col·laborativament amb els responsables de certs mòduls del codi, per tal de poder dur a terme la integració amb major coneixement de domini.

Aquest segon període compren les fases 2 i 3 (Secció 8)



## 4. Planificació Temporal

### 4.1 Calendari

El projecte comença el 3 de febrer i acaba el 30 de juny. La defensa es realitzarà el 30 de juny a les 11:00 AM, i la memòria s'ha d'entregar el 23 de juny.

Durant la durada del projecte es cursarà l'assignatura GEP, del 17 de febrer al 14 de març.

### 4.2 Descomposició de Tasques

#### 4.2.1 Fases del projecte

El projecte es divideix en 4 fases principals o iteracions

**Pre-Fase o Fase 0:** L'objectiu d'aquesta iteració és definir l'abast del projecte. A partir d'un seguit de reunions es du a terme un estudi i es prepara l'entorn de treball.

La duració d'aquesta fase és de 3 setmanes.

**Fase 1:** L'objectiu principal d'aquesta fase és el de definir els requisits del projecte i fer una primera aproximació del disseny de l'aplicació. Les tasques principals d'aquesta fase són les següents:

- Elicitar i negociar els requisits del projecte
- Especificar i documentar aquests requisits
- Proposar una primera aproximació de l'arquitectura del software, on s'identifiquen els mòduls principals que cal implementar. Això inclou el control de flux i els repositoris de metadades.

Volem aconseguir una bona base per a començar a treballar, de forma que ens puguem anticipar a possibles riscos. Hem de documentar amb prou detall per tal de poder prevenir els problemes que puguin aparèixer durant la fase 2.

En paral·lel, cal començar a familiaritzar-se amb les llibreries i tecnologies necessàries per a dur a terme el projecte: Servidors, llenguatges, llibreries gràfiques...

La durada d'aquesta fase serà al voltant de 3 setmanes (fins a finalitzar GEP).

**Fase 2:** Aquesta fase consisteix principalment en la implementació del projecte. Aplicarem la metodologia de *Desenvolupament de Software Agile*. Descomposarem les tasques identificades a la iteració prèvia amb una granularitat més fina.

Realitzarem tasques de test a partir de prototips i demos tècniques. La durada d'aquesta iteració serà d'aproximadament 10 setmanes.

**Fase 3:** Finalment en aquesta última fase integrarem totes les peces i construirem el software. Caldrà revisar la feina feta, tancar el projecte i documentar.

La durada d'aquesta iteració serà de 2 setmanes.

## 4.2.2 Tasques Principals del projecte

A continuació definirem les tasques que hem identificat per cadascuna de les 4 iteracions. Podem veure una estimació del temps que ens portarà cadascuna a la Secció 4.2.2.5.

### 4.2.2.1 Tasques de la Fase 0

**Posada en marxa de l'entorn de treball:** Hem de preparar el nostre entorn de treball. Identifiquem les següents subtasques:

- Obtenir el software: Cal descarregar i preparar tot el software necessari per treballar, així com els *Software Development Kits* (SDKs) i llibreries necessàries.
- Incorporar el codi llegat: Donat que treballarem amb una gran quantitat de codi de diferents fonts i llenguatges, necessitem comprovar que tot funciona i incorporar-ho al nostre projecte.

**Antecedents:** Cal comprendre el codi dels mòduls amb els que treballarem. Això inclou:

- Familiaritzar-se amb el context: Primerament cal llegir i aprendre sobre el data warehousing, els processos ETL i altres conceptes contextuais que s'utilitzaran durant el projecte.
- Lectura de documentació: Per tal de comprendre com funcionen els mòduls del projecte, cal llegir un seguit de documentació i papers sobre aquests.
- Revisió del codi: Un cop hem entès com funcionen els mòduls existents, cal estudiar el codi per veure com han estat implementats i com funcionen per dins, ja que haurem de modificar-ne algunes classes i adaptar-los al sistema general.

**Planificació inicial i descomposició de tasques:** Cal definir un pla de treball inicial, identificar les tasques principals, fer una primera descomposició i una estimació en hores de la duració d'aquestes. Aquesta planificació serà modificada i adaptada a mesura que el projecte avanci (ja que estem utilitzant una metodologia *Agile*) però ens cal disposar d'aquesta base sobre la que treballar.

### 4.2.2.2 Tasques de la Fase 1

**Recerca:** La primera tasca implica recerca. La recerca es farà de forma transversal, no només durant aquesta iteració sinó durant tot el projecte, però ens cal un treball més intens durant el transcurs de la primera iteració. Cal investigar quines tècniques de visualització i interacció amb grafs s'utilitzen actualment, i amb quins llenguatges i tecnologies funcionen. Cal determinar quines eines s'adapten millor al nostre ecosistema. També cal veure com de flexibles són, i advertir com de difícils serà adaptar-les al nostre projecte i implementar les funcionalitats que desitgem.

**Entrenament:** Necessitem realitzar entrenament amb aquelles eines software o APIs que vulguem utilitzar.

- Entrenament en diagrames BPMN: Ens cal aprendre com dissenyar diagrames BPMN. Per a aquesta tasca utilitzarem Microsoft Visio, de forma que haurem d'aprendre a utilitzar aquest software.
- Entrenament en la eina de visualització de grafs interactius. Començarem creant exemples bàsics amb interaccions simples de forma que puguem familiaritzar-nos

amb l'eina a la vegada que aprenem a utilitzar JavaScript i les llibreries gràfiques. Concretament ens centrarem en d3.js, que en aquest moment és la gran candidata, ja que és de les poques llibreries que permeten a l'usuari un cert grau d'interactivitat.

- **Entrenament en JavaScript i desenvolupament d'aplicacions web:** Aquesta tasca no es trobava a la llista inicial de tasques, ja que inicialment no teníem pensat implementar una aplicació web, però durant la iteració 0 vam prendre aquesta decisió arrel d'un seguit de factors que trobem documentats als informes a de la secció 6.1. Ens caldrà aprendre a utilitzar el llenguatge JavaScript, així com tot allò que se'ns requereixi per a desenvolupament web (HTML, CSS, servidors...).

**Documentació:** Donat que aquesta iteració coexistirà amb GEP, gran part de la documentació generada per aquesta assignatura apareixerà també a la memòria final del TFG. També generarem un conjunt de documents com els diagrames BPMN dels processos que tinguem definits, *mockups* de les interfícies, mapes navegacionals, entre d'altres.

**Planificació:** Durant el transcurs de la iteració caldrà revisar la planificació inicial.

#### 4.2.2.3 Tasques de la Fase 2

**Visualització d'ontologies a partir de la API triada:** Les ontologies són el recurs principal per tal de que l'usuari del sistema pugui definir requisits. Així, la primera tasca serà aconseguir representar-les visualment com a grafs. Caldrà poder llegir fitxers .OWL (és l'estàndard del W3C per a representar ontologies) i interpretar-los gràficament.

**Interacció de l'usuari amb el graf:** El següent pas es implementar totes les interaccions possibles de l'usuari amb el graf: Esdeveniments de clic amb el ratolí, menús contextuais, accions de *drag & drop*, etcètera. El propòsit d'aquesta interacció és el de definir els requisits de negoci. Cal notar que el resultat de la interacció és un graf anotat representant un requisit de negoci.

**Creació dels requisits de negoci:** Un cop l'usuari ha dissenyat el seu requisit de negoci, cal reinterpretar el graf i obtenir un fitxer que el procés principal (GEM+ORE+COAL) sigui capaç d'entendre. Cal crear uns pàrsers que ens permetin fer la transformació, utilitzant documents JSON i XML.

**Execució del procés i emmagatzemament de resultats:** A la banda del servidor tindrem les classes que processen els requisits i els transformen en fitxers de sortida. Depenent de l'estat d'alguns mòduls que estan essent desenvolupats en altres projectes podrem obtenir més o menys quantitat de fitxers de sortida. Haurem de generar aquest fitxers i guardar-los al repositori per tal que es puguin obrir amb el visor de resultats.

**Interfície gràfica:** Aquesta tasca es independent a la resta. Es tracta de dissenyar i implementar una interfície que permeti a l'usuari navegar per tot el software i connecti tots els mòduls. Amb la interfície permetrem crear, carregar i esborrar projectes, llençar els diferents processos, obrir l'editor de requisits...

**Disseny i implementació del repositori del Minecart:** De forma transversal a la implementació del editor de requisits, caldrà dissenyar la base de dades interna i implementar les APIs de

comunicació i els pàrsers de dades. Aquesta tasca s'anirà completant a mesura que es necessitin els recursos.

#### 4.2.2.4 Tasques de la Fase 3

L'objectiu principal d'aquesta iteració és agrupar totes les peces, produir el software i tancar el projecte. També caldrà generar la documentació final.

#### 4.2.2.5 Estimacions Temporals

Una primera aproximació a la càrrega de treball que ens portarà cada fase del projecte, donat que aquest consta de 18 crèdits ECTS, és la següent:

Fase 0: 80 hores

Fase 1: 150 hores

Fase 2: 260 hores

Fase 3: 60 hores

#### 4.2.2.6 Consideracions

Com s'ha comentat diverses vegades durant aquest documents, algunes de les tasques descrites requereixen l'ús de tecnologies amb les que no ens trobem plenament familiaritzats (per exemple, el desenvolupament d'aplicacions web). Es possible que les estimacions i la descomposició de tasques siguin incorrectes, de forma que aquesta aproximació inicial queda subjecta a canvis. Aquest és, de fet, un dels motius per l'implantació *d'Agile*: el grau d'incertesa respecte certs aspectes del projecte.

## 4.3 Diagrama de Gantt inicial

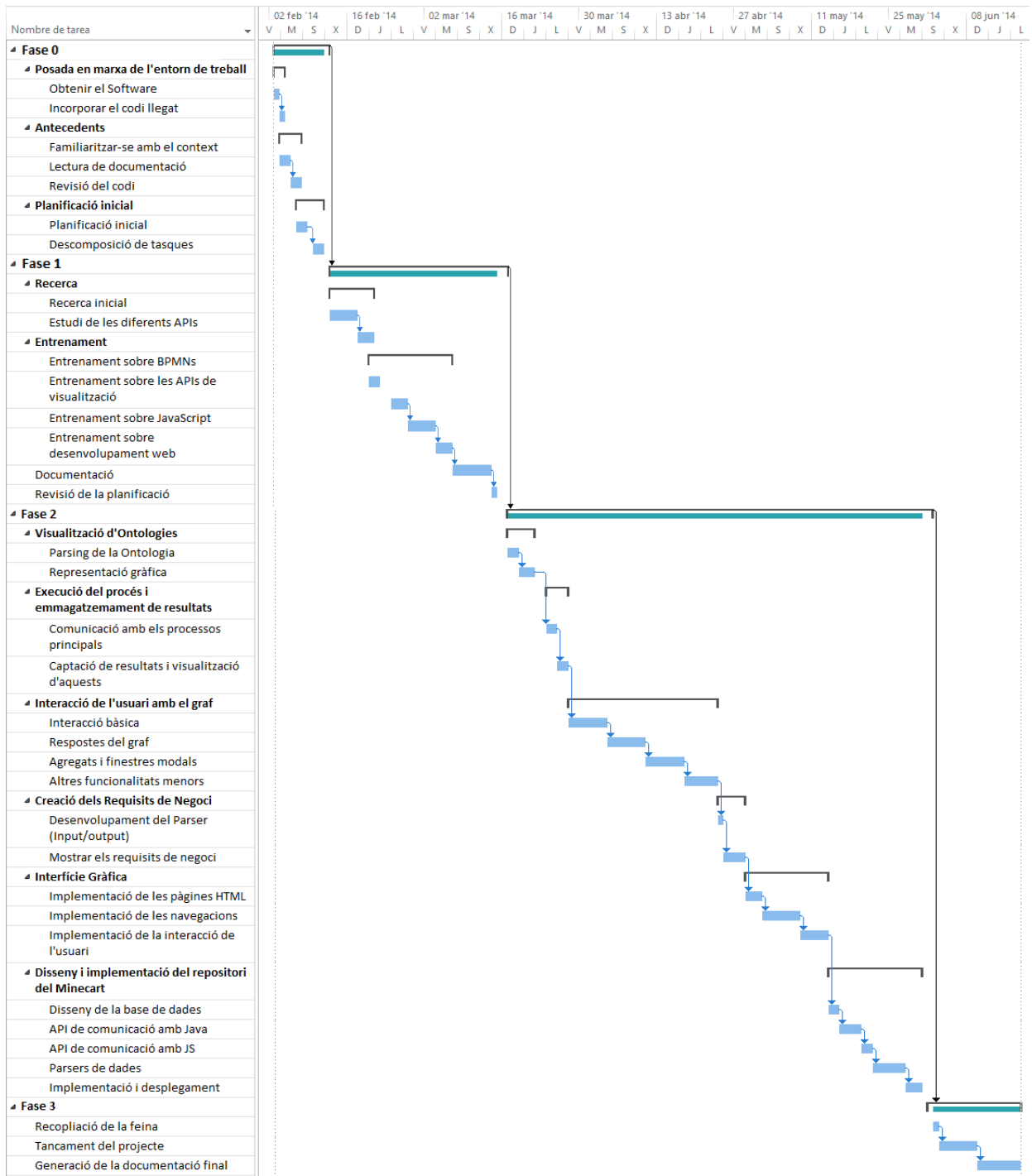


Figura 10: Diagrama de Gantt inicial del projecte [Font: propi]

## 4.4 Control de Desviacions

Com s'ha discutit a la secció 2.3, hi ha un gran nombre de factors en aquest projecte que poden derivar en desviacions a la planificació. Per tal de reaccionar-hi de forma eficient definim les següents estratègies:

- Marge de maniobra: Com podem veure al diagrama de Gantt, hem planificat les tasques per tal d'acabar el projecte el 15 de juny, però la memòria del projecte s'ha d'entregar el 23 de juliol i la lectura és el dia 30 de juliol. Això ens deixa uns dies de marge per tal de poder maniobrar i redefinir la planificació si algunes tasques requereixen molt més temps del que vam estimar, o si ens trobéssim en indisposició alguns dies.
- Revisions setmanals: Un dels avantatges *d'Agile* és que implica celebrar reunions regulars (en el nostre cas setmanals), de forma que el projecte es defineix a si mateix a mesura que es desenvolupa. Això significa que estarem modificant i millorant contínuament la planificació inicial, de forma que ens permet reaccionar amb molta antelació davant desviacions, i redirigir el nostre treball per tal d'assolir l'objectiu final.
- La descomposició inicial de tasques és prou fina per poder veure si algunes tasques ens estan requerint més temps de l'esperat o no.

En resum, si seguim estrictament la nostra metodologia i detectem i corregim les desviacions a temps, el nostre projecte no patirà cap entrebanc i el treball es finalitzarà a les dates acordades.

## 5. Pressupost i Sostenibilitat

### 5.1 Identificació i estimació de costs

#### 5.1.1 Recursos Humans

Aquest projecte forma part d'un projecte més gran (secció 1.2) i serà desenvolupat per una sola persona com a part del Treball de Final de Grau (tot i que es comptarà amb el suport del codirector del projecte per a alguns afers tècnics). Tenint això en compte, i seguint la planificació especificada a la secció 3, podem veure a la Taula 1 els costos associats als recursos humans.

Hem detallat el cost de cada iteració, per a poder monitoritzar millor el pressupost, i hem marcat un preu per hora de 8€, ja que és el preu que es paga als estudiants que estan fent conveni empresa-universitat a través d'un contracte amb UPC.

Iteració	Hores Estimades	Preu per Hora	Cost Total
Fase 0	80	8€/h	640 €
Fase 1	120	8€/h	960 €
Fase 2	300	8€/h	2.400 €
Fase 3	50	8€/h	400 €
Total	550	-	4.400,00 €

Taula 1: Pressupost dels recursos humans

#### 5.1.2 Hardware

A la taula 2 mostrem la llista de hardware necessari per a dur a terme el projecte. Detallem també el preu del hardware, la vida útil, i una amortització estimada basada en la duració total del projecte<sup>1</sup>.

El camp de vida útil només es una aproximació basada en experiències prèvies, hem de tenir en compte que pot no ser exacte.

Ja disposàvem de tots els elements hardware necessaris per al projecte, de forma que els preus que apareixen a les taules son referents a quan es van adquirir, i no pas al seu preu actual al mercat.

Hardware	Preu	Vida Útil	Amortització Estimada
Lenovo IdeaPad Y580	980€	4 anys	101.92€
Monitor BenQ XL2420TX	400€	4 anys	41.60€
Total	1,478€	-	143.52€

Taula 2: Pressupost dels recursos hardware

<sup>1</sup> Assumint que la duració del projecte és de 5 mesos, tal i com es constata a la secció 4.1

### 5.1.3 Software

Durant el transcurs del projecte necessitarem utilitzar tot tipus de software. Gran part del software l'hem adquirit sota llicències educacionals, de forma que no comporten cap cost per a nosaltres. Un altre gran part de software (llibries i eines) que hem utilitzat també es troba sota llicència *Open Source*, de forma que tampoc comporta cost.

Pel control de versions i per compartir documents utilitzarem un repositori de *subversion* anomenat *TortoiseSVN*[18], que és també gratuït.

Finalment utilitzarem JIRA[19] per a per al control de bugs/problemes i per al *product backlog*. JIRA és un producte comercial, però també hem aconseguit adquirir-lo amb una llicència educativa a través de la Universitat.

Hardware	Preu	Vida Útil	Amortització Estimada
Eclipse IDE for Java EE Developers	0	-	0 €
Eclipse JavaScript Development Tools	0	-	0 €
Microsoft Office 365 Small Business Premium	12.80€ Usuari/mes	-	64 €
Microsoft Visio 2013	0€ <sup>2</sup>	-	0 €
Microsoft Project 2013	0€ <sup>2</sup>	-	0 €
Tortoise SVN	0	-	0 €
Jira License	0€ <sup>2</sup>	-	0 €
Notepad++	0	-	0 €
Apache Tomcat	0	-	0 €
Balsamiq Mockups	57.8€	2 anys	12.02€
Microsoft Windows 8 Pro	145.61€	3 anys	Amortitzat amb el Lenovo IdeaPad Y580
Total	267.41€		86.02€

Taula 3: Pressupost dels recursos software

### 5.1.4 Despeses Generals

Les següents despeses son aquelles relacionades amb la companyia: El preu del local on es treballa i les seves despeses associades:

Concepte	Cost Estimad
Lloguer d'espai de treball a l'entorn col·laboratiu Itnig	199€/mes
Llum, aigua i gas del local	Inclòs al lloguer de l'espai
Telèfon	20€/mes
Llicències	0€ <sup>3</sup>
Total	1.095,00 €

Taula 4: Pressupost de les despeses generals

<sup>2</sup> El preu és de 0€ perquè, com hem explicat, hem obtingut aquest software a partir d'una llicència educativa gratuïta.

<sup>3</sup> No hem de pagar per llicències de treball ni d'autònom perquè treballem amb un contracte d'estudiant amb la UPC.



### 5.1.5 Pressupost Total

Concepte	Cost Estimat
Recursos Humans	4,400.00€
Hardware	143.52€
Software	86.02€
Despeses Generals	1.095 €
Total	5,725.54€

Taula 5: Pressupost Total

### 5.1.6 Viabilitat Econòmica

Amb aquests números podem assegurar la viabilitat del projecte. Hem comentat ja diversos cops durant aquest document que utilitzarem la metodologia de Desenvolupament de Software *Agile* (ASD). Això implica que els requisits no es troben estrictament definits al principi del projecte, de forma que el procés de desenvolupament ha de ser capaç de tenir en compte noves necessitats i canvis que puguin anar apareixent.

Quan es treballa amb la clàssica metodologia “en cascada” existeix el que s’anomena el *iron triangle* (Figura 12). El *iron triangle* considera que un projecte té tres variables principals: els requisits o funcionalitats, el cost i la planificació, i ens diu que fixant la variable dels requisits podem arribar a estimar i fixar les variables de cost i planificació.

*Agile* ens permet fixar les variables que nosaltres considerem necessàries i deixar-ne d’altres obertes a canvis o modificacions durant el transcurs del projecte. Nosaltres només podem fixar en aquest moment la variable de la data d’entrega i alguns aspectes de la planificació. Així, la resta serà flexible.

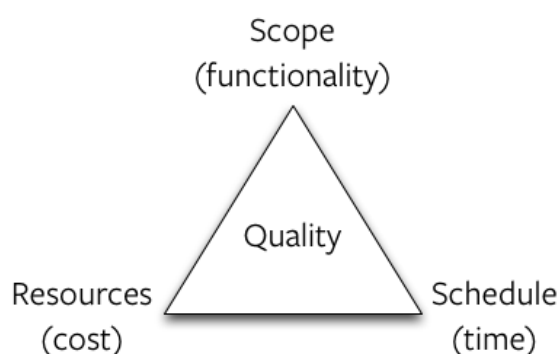


Figura 11: Representació gràfica del Iron Triangle clàssic de la metodologia “en cascada” [20]

Com es tracta d’un projecte de recerca, estarem en tracte constant amb el client, que és qui s’encarrega de portar el projecte GEM. Així, com el pressupost inicial està clarament definit i aprovat, la durada del projecte és fixa i els requisits s’aniran constituint amb un contacte continu amb el client i no suposaran una variació a la durada o al cost, podem assumir que el projecte és viable.

## 5.2 Control de Pressupost

El procés de desenvolupament de software *Agile* també ha de considerar possibles ajustaments en el temps i els recursos dedicats a les diferents tasques del projecte. Pot passar que sorgeixi la necessitat de nou hardware o software per a finalitzar el projecte, per exemple. És per aquest motiu que un control de pressupost eficient és necessari.

Si això succeís seria sempre arrel d'un acord amb el client. Aquests possibles canvis i modificacions que comportin nous costs seran producte de les reunions del final de cada *sprint*, de forma que el client sempre sabrà la repercussió en el pressupost que comportaria realitzar aquest canvi, i sempre es realitzaran sota el seu vistiplau.

Documentarem en detall tota la càrrega de treball i les hores invertides en cada tasca, de forma que utilitzant aquesta informació i la planificació inicial podrem modificar el pressupost si és necessari i mostrar-ho al client per tal de que pugui prendre una decisió.

Totes les desviacions que apareguin sobre el pressupost inicial seran documentades.

## 5.3 Impacte

### 5.3.1 Impacte Social

Avui dia el disseny i la implementació d'un data warehouse és una tasca manual que requereix de professionals especialitzats i consumeix una gran quantitat de temps.

**Automatitzant** aquest procés volem aconseguir el següent:

- Alliberarem de forma parcial els especialistes en Data Warehousing que es troben al càrrec d'aquesta tasca. Així, poden formar part d'altres projectes o tasques que puguin requerir la seva expertesa.
- Permetrem que usuaris no-especialitzats puguin dissenyar, editar i utilitzar el seu propi sistema de data warehouse. Aquest concepte de democratitzar l'accés a l'anàlisi de dades per als usuaris no experts es coneix com a *anàlisi de dades per a masses*[21].
- L'ús del software millorarà el funcionament general i la gestió de projectes de les empreses.

A més a més, el sistema resultant seguirà la arquitectura tradicional d'una aplicació web, és a dir, tindrà el costat del servidor i el del client separats. El procés principal (GEM+ORE+COAL) correrà sobre el servidor, de forma que eventualment es podrà desplegar sobre el núvol i es podrà distribuir com a servei. Així, serà possible per a petites i mitjanes empreses començar els seus projectes amb data warehouses utilitzant aquest servei, reduint els costs inicials i evadint el cost de contractar un expert en data warehouses. Actualment, i d'acord amb una enquesta de l'empresa Gartner [22], el 60% dels projectes data warehousing es retarden i en general, el 90% triguen més d'un any en entrar en funcionament.

### 5.3.2 Impacte Mediambiental

El nostre projecte contribuirà en l'esforç de conservació d'energia:

- Com hem constatat al punt anterior, el procés de creació del data warehouse serà quasi automatitzat. Això implica una reducció de l'ús dels recursos i del personal, i en conseqüència un estalvi d'energia (portàtils, computadores, calefacció, llum, electricitat...).
- Desplegar el sistema al núvol optimitzarà l'ús de recursos i ens permetrà noves opcions d'estalvi d'energia. *Cloud computing* segueix la premisa del consum elàstic (consum segons les necessitats)

## 6. Incepció

En aquesta secció trobem tota la documentació relacionada amb la Fase 1 del projecte. Aquesta fase 1 correspon amb el primer *sprint* que es realitza seguint la metodologia *Agile*, que s'anomena *inception*, o Iteració 0. És un *sprint* relativament diferent a la resta, pels següents motius:

- Tindrà una durada de 6 setmanes, del dia 12 de Febrer al 18 de Febrer.
- L'objectiu d'aquest *sprint* és començar a definir el projecte i elaborar una elicitació de requisits inicial, de forma que puguem crear la llista d'*user stories* que conformen el Product Backlog inicial amb el que començarem a treballar a la següent fase.
- Cada setmana es realitza una reunió presencial amb el director i el codirector del projecte. S'avaluen els avanços, es posen en comú les tasques fetes durant la setmana i utilitzant tècniques de *brainstorming* es defineixen les bases del projecte.
- De cada reunió se'n deriva un document on consta l'objectiu de la reunió, un resum d'aquesta i el seguit de tasques a realitzar per a la següent reunió.

A continuació mostrem un resum dels informes de les reunions presencials per a poder resumir les tasques de la fase, el document d'elicitació de requisits i el Product Backlog Inicial.

### 6.1. Informes

#### Setmana 1

Data: 12/02/2014

**Objectiu de la Reunió:** L'estudiant ha començat a llegir documentació i papers sobre el projecte, i ja s'hi troba en contacte de forma que cal fer una primera reunió per tal de definir bé la metodologia que s'utilitzarà, parlar de l'abast del projecte i definir alguns punts relacionats amb l'arquitectura i les funcionalitats.

**Resum de la Reunió:** S'ha parlat en profunditat sobre l'abast del projecte, i ara es té una visió global. El desenvolupament es durà a terme en dues fases. La primera fase tindrà un treball d'especificació de requisits i a la segona fase utilitzarem una metodologia *Agile* basada en iteracions setmanals. S'han acordat reunions setmanals, tant per a la primera fase com a la segona, tots els dimarts a les 3:30 PM.

També s'ha decidit canviar la estructura del projecte. Es desenvoluparà una aplicació web utilitzant JavaScript a la banda del client i Java a la banda del servidor. AMDO, GEM, COAL i ORE es trobaran a la banda del servidor, i la interfície serà totalment reconstruïda a la banda del client utilitzant HTML, CSS i JavaScript. El graf interactiu, la visualització de resultats i tot allò relacionat amb la GUI es trobarà també a la banda del client.

Per últim s'ha decidit que les dades internes es guardaran en una base de dades NoSQL (una base de dades clau-valor anomenada mongoDB[23]). El motiu és que generem documents i aquesta és una base de dades de documents que a més escala fàcilment i dona bon rendiment. Aquesta base de dades es col·locarà al servidor i serà accessible des d'ambdues bandes.

#### Tasques:

- Formalitzar l'arquitectura del projecte.
- Començar a escriure els requisits funcionals del projecte i desenvolupar-los utilitzant diagrames BPMN o *Mockups*.
- Posar-se en contacte amb la metodologia *Agile* i escriure un document resumint l'abast del projecte.

## Setmana 2

Data: 18/02/2014

**Objectiu de la Reunió:** L'estudiant ha elaborat els documents que es van demanar en l'última reunió. Aquests documents són: Un esquema del projecte, una llista inicial de requisits i un conjunt de *mockups* que representen les funcionalitats del graf interactiu. L'objectiu d'aquesta reunió és discutir sobre aquests.

**Resum de la Reunió:** L'estudiant ha mostrat la figura que representa l'arquitectura del projecte. Aquesta figura resulta molt útil ja que ens ajuda a tenir un grau d'abstracció i poder veure el projecte des de fora.

Seguidament l'estudiant mostra un esbós del document que descriu els requisits funcionals. Es discuteix sobre la correctesa d'aquests i es conclou que s'ha de treballar més a fons. Cal expandir la llista, i ser una mica més formals a les definicions.

Finalment l'estudiant explica com creu que hauria de ser la interfície gràfica i quines interaccions hauria de suportar.

#### Tasques:

- Elaborar els dos documents que es requereixen per a les entregues de GEP: L'abast i la metodologia del projecte i la descomposició de tasques.

## Setmana 3

Data: 25/02/2014

**Objectiu de la Reunió:** L'objectiu d'aquesta reunió és discutir sobre els dos primers entregables de GEP. Llegirem el feedback que ha donat el tutor de GEP i parlarem de les millores que s'han de fer en cara a la documentació final.

**Resum de la Reunió:** El principal problema és que, al estar utilitzant la metodologia de Desenvolupament de Software *Agile* (ASD), és difícil definir aquests punts al principi del projecte. També s'ha seguit parlant i redefinint el document dels requisits.

#### Tasques:

- Cal preparar el document de Sostenibilitat i Pressupost
- Cal preparar el document que defineix l'Estat de l'Art
- Cal començar a plantejar els requisits funcionals del projecte

## Setmana 4

Data: 04/03/2014

**Objectiu de la Reunió:** L'objectiu d'aquesta reunió és assaltar el document de especificació de Requisits Funcionals. Ara que ja estem en contacte amb el projecte i hem definit els bàsics, cal revisar i validar aquesta primera definició de requisits.

**Resum de la Reunió:** S'han llegit els documents un per un i s'ha afegit/eliminat el que feia falta. S'ha decidit com han de ser documentats:

- Els requisits queden dividits en dos grups: Relacionats amb la interacció entre mòduls i relacionats amb el MineDigger i el visor de resultats.
- Com MineDigger i el visor de resultats no son prou clars encara, només en farem una breu descripció i els il·lustrarem amb Mockups.
- Els requisits de la interacció entre mòduls son molt clars en aquest punt, de forma que proveirem una descripció més completa, un diagrama BPMN de flux i un mapa navegacional.

### Tasques:

- Preparar la presentació prèvia de GEP i corregir i acabar el document de Requisits Funcionals.

## Setmana 5

Data: 11/03/2014

**Objectiu de la Reunió:** GEP ha acabat, i cal començar a planificar la segona fase del projecte.

**Resum de la Reunió:** Aquesta reunió ha sigut molt breu. Hem començat a parlar de les llibreries gràfiques que utilitzarem per a desenvolupar el MineDigger i hem començat a planificar les tasques i dates per a la segona fase del projecte.

### Tasques:

- Entregar la documentació final de GEP, corregida, i preparar la presentació oral.
- Començar a treballar amb les llibreries gràfiques i elaborar un anàlisi.
- Començar a treballar amb JavaScript i Node.js
- Replanificar el projecte.

## Setmana 6

Data: 18/03/2014

**Objectiu de la Reunió:** Aquesta és l'última reunió de l'*sprint* 0. Es tracta d'un punt d'inflexió. Cal revisar la planificació i tancar la fase del projecte.

**Resum de la Reunió:** S'ha fet un últim cop d'ull a l'especificació de requisits inicial i s'ha revisat que tot fos correcte. L'estudiant ha estat treballant una mica amb les eines i s'han estat mirant exemples de funcionament per tal de valorar si podrem dur a terme el projecte amb ella.

### Tasques:

- Fer una llista de les funcionalitats que necessitem de la llibreria gràfica i veure si estan suportades.
- Triar la millor opció per a implementar l'arquitectura client-servidor i fer el *training*.
- Preparar el primer *sprint*.

## 6.2 Elicitació de Requisits

En aquesta secció mostrem la **primera aproximació** dels requisits funcionals de l'aplicació. Aquest document ha estat elaborat durant tot el transcurs de l'*sprint* 0. Cal entendre aquesta secció com una **primera elicitació que patirà canvis i que no reflexa el resultat final**. Podem veure l'evolució d'aquests requisits en la documentació dels *sprints* (Secció 6), i el resultat final al *Product Backlog* resultant (Secció 6.14) i a la descripció de l'eina (Secció 7)

La nostra aplicació tindrà quatre components principals:

1. Interfície Principal del Sistema (*Main Sysyem Interface* - MSI).
2. Un editor de requisits de negoci (MineDigger) al que accedirem des de la MSI.
3. Finestra de resultats (*Output Screen* - OS) a la que accedirem des de la MSI.
4. Un repositori intern que s'encarregui de la interacció entre els mòduls i el control de flux.

En el moment en que es redacta aquest document alguns dels requisits disposen d'un major nivell de detall que d'altres. Així, hem decidit fer aquest primera elicitació donant un major o menor nivell de detall als requisits depenent de a quin component pertanyen. De nou, estem seguint els principis de la metodologia *Agile*. En el transcurs dels *sprints* els anirem redefinint i ampliant a través de les *user stories*. Així:

1. Els requisits de la Interfície Principal del Sistema (Secció 6.2.2.1) es documenten com segueix:
  - Definició detallada del requisit.
  - Pre i post condicions.
  - Diagrama de negoci BPMN mostrant la comunicació entre els components del Sistema i els seus usuaris.

També incloem un mapa navegacional de la interfície.

2. Els requisits del MineDigger (Seccions 6.2.2.2) es documenten com segueix:
  - Definició detallada del requisit.
  - Pre i post condicions.

També incloem un seguit de *mockups* mostrant l'aparença que tindrà el MineDigger a la implementació final, i com s'utilitzarà en relació a la definició de requisits.

3. Els requisits de la finestra de resultats (Secció 6.2.2.3) es documenten com segueix:
  - Definició detallada del requisit.
  - Pre i post condicions.
4. Finalment els requisits que pertanyen al repositori intern, degut a la naturalesa d'aquest component, es documentaran de forma transversal sobre els requisits dels tres altres components.

## 6.2.1 Vocabulari

Abans de començar amb la definició dels requisits, ens agradaria fixar el vocabulari del domini.

- **Concepte/Node:** Un concepte és un element d'una ontologia. Al MineDigger carregarem ontologies i podrem visualitzar-ne els seus conceptes com nodes d'un graf, de forma que utilitzarem aquests dos mots (concepte i node) de forma indiferent.
- **Rol:** El rol multidimensional d'un concepte pot ser *Fet*, *Dimensió* o *Mesura*. El rol multidimensional d'un concepte defineix si aquest és un focus d'anàlisi (*fet*), la perspectiva des de la qual el fet és analitzat (*dimensió*) o un atribut d'aquest (*mesura*).
- **Concepte Rellevant:** L'usuari pot marcar alguns conceptes com a rellevants per a ell. Aquests seran els que tindrem en compte quan construïm el requisit de negoci.
- **Pantalles:** La interfície estarà dividida en diferents pantalles. Tenim la *pantalla principal*, on podem crear, carregar i esborrar projectes a les pantalles de *creació*, *càrrega i esborrat de projectes* respectivament. Quan un projecte s'obre podem fer un seguit d'accions a la *pantalla d'administració del projecte*.
- **Col·leccions i Documents:** MongoDB és una base de dades no relacional que es divideix en col·leccions i documents. Podríem considerar que son les *taules* i *files* d'una base de dades relacionals, tot i que son mols més flexibles i *schemaless*.

## 6.2.2 Requisits

### 6.2.2.1 Interfície Principal del Sistema

Aquest conjunt de requisits conté tots aquells requisits relacionats amb la interfície principal del sistema i defineix les diferents navegacions que podem fer. Els diagrames BPMN es troben en anglès degut a que son materials que es van generar pensant en futurs projectes en els quals la llengua vehicular dels seus membres és l'anglès, i hem preferit mantenir els fitxers originals a la documentació.

#### Requisit #001: Crear un projecte nou

**Descripció:** En primer lloc l'usuari comunica al Sistema que vol crear un projecte nou. Seguidament el Sistema mostra un formulari on l'usuari haurà d'emplenar un conjunt de camps. Aquests camps son els següents:



- Nom del projecte: Una cadena de 40 caràcters com a màxim. No estan permesos els caràcters especials (per incompatibilitats amb l'estructura del sistema d'emmagatzematge de dades).
- Ontologia associada: Un fitxer d'extensió .OWL que descriu una ontologia. Utilitzarem una finestra d'exploració d'arxius per a carregar-lo.
- Fitxer de mapeig de les fonts (*Source Mappings*): Un fitxer XML que descriu el mapeig de les diferents fonts amb els conceptes de la ontologia per al projecte. Utilitzarem una finestra d'exploració d'arxius per a carregar-lo.

Aquestes dades seran parsejades en diferents formats i emmagatzemades internament al repositori intern, en una col·lecció que s'encarregarà d'emmagatzemar les metadades.

Un cop finalitzat el procés, el sistema mostra a l'usuari la pantalla d'administració del projecte.

**Pre-condició:** Ens trobem a la finestra principal.

**Post-condició:** El projecte s'ha creat. Les metadades del projecte s'han emmagatzemat al Minecart. Es mostra la finestra de gestió del projecte.

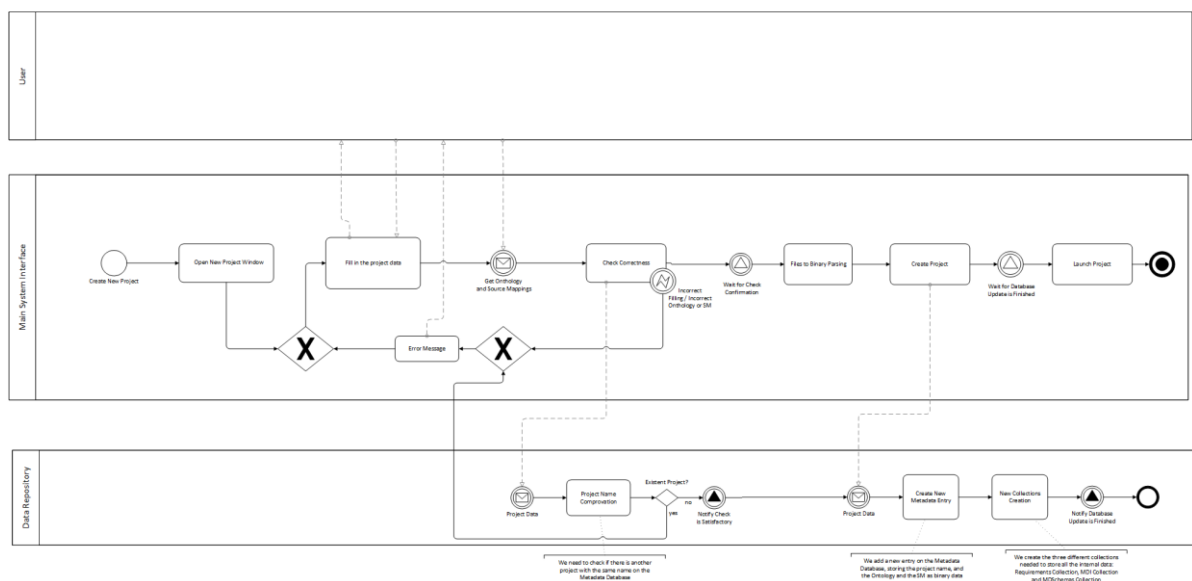


Figura 12: Diagrama BPMN del requisit #001 [Font: Propi]

## Requisit #002: Obrir un projecte existent

**Descripció:** L'usuari indica al sistema que vol obrir un projecte existent. El sistema mostra una llista desplegable amb el nom de tots els projectes emmagatzemats al sistema. Aquesta informació es recull de la col·lecció de metadades del repositori.

Així, l'usuari tria un projecte i el sistema demana certa informació sobre el projecte al repositori de dades. Aquesta informació és la següent:

- El nombre de requisits de negoci que s'han fet al projecte.
- Els fitxers de sortida que s'han generat i guardat prèviament en aquell projecte.

**Pre-condició:** L'usuari es troba a la finestra principal. Com a mínim existeix un projecte al sistema.

**Post-condició:** Es mostra la finestra de gestió del projecte que l'usuari ha obert.

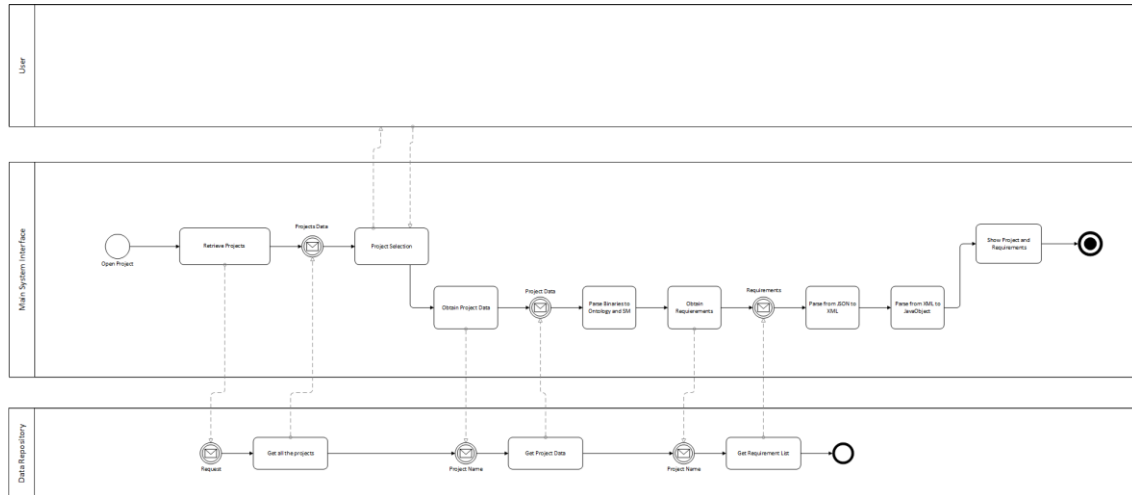


Figura 13: Diagrama BPMN del requisit #002 [Font: Propi]

### Requisit #003: Esborrar un projecte existent

**Descripció:** L'usuari li indica al sistema que vol esborrar un projecte existent. El sistema mostra una llista desplegable amb els noms dels projectes emmagatzemats al sistema. Aquesta informació es recull de la col·lecció de metadades del repositori de dades.

Seguidament, l'usuari tria un projecte. El sistema informa a l'usuari que s'esborraran totes les dades d'aquest projecte. Un cop l'usuari confirma, el sistema esborra la informació d'aquest projecte de la col·lecció de Metadades. També esborra els requisits de negoci que s'hagin creat per aquell projecte (si existeixen), els fitxers de resultats que hem guardat (si existeixen) i dades internes relacionades com els continguts de les col·leccions de grafs multidimensionals i ETLs associades al projecte.

Finalment el sistema mostra de nou la finestra principal.

**Pre-condició:** L'usuari es troba a la finestra principal. Al sistema hi ha com a mínim un projecte.

**Post-condició:** El projecte triat per l'usuari i tota la informació associada son esborrats del sistema.

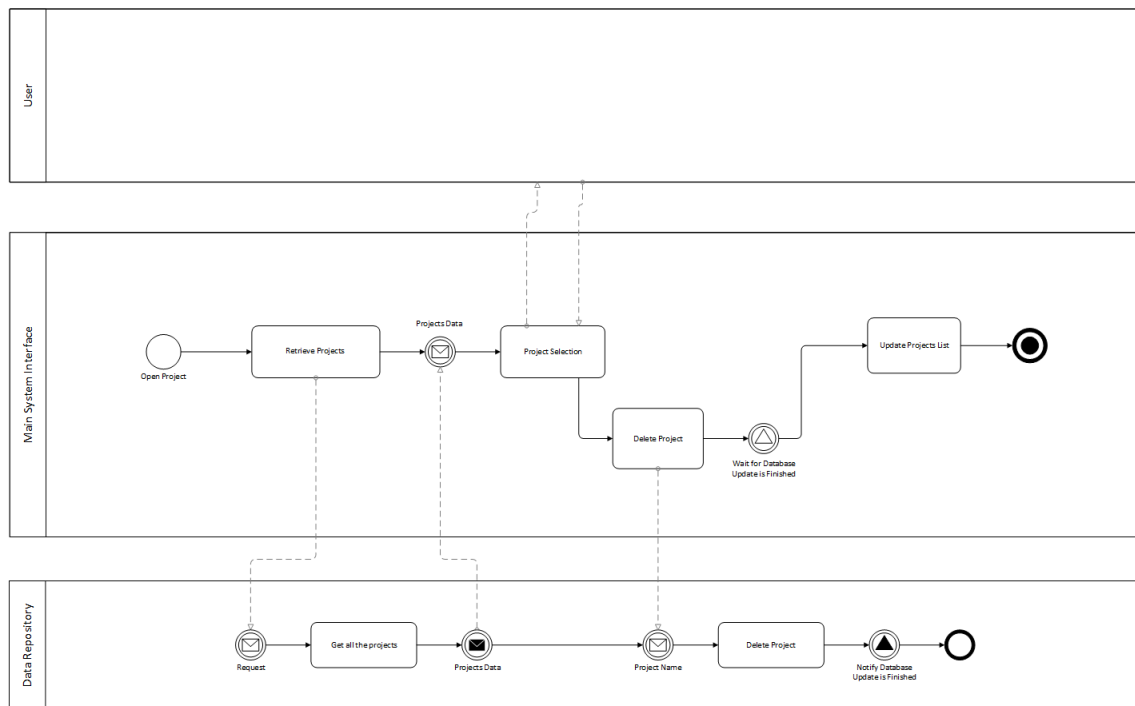


Figura 14: Diagrama BPMN del requisit #003 [Font: Propi]

#### Requisit #004: Canviar Ontologia

**Descripció:** L'usuari indica al sistema que vol canviar l'ontologia del projecte que es troba obert. El sistema adverteix a l'usuari que s'esborraran totes les dades associades al projecte exceptuant el nom del projecte. Seguidament l'usuari carrega la nova Ontologia i el sistema la guarda a la col·lecció de Metadades del repositori.

**Efecte Col·lateral:** Un cop la ontologia és canviada, tots els requisits de negoci i els fitxers de resultats del projecte han de ser invalidats per assegurar la consistència del sistema. Ara per ara el sistema esborrarà els fitxers del repositori de dades, però en futurs projectes es podrà dissenyar un mòdul per readaptar els requisits existents a la nova ontologia.

**Pre-condició:** Hem obert un projecte i l'usuari es troba a la pantalla de gestió de projectes.

**Post-condició:** La ontologia associada al projecte obert és esborrada i una nova ontologia és carregada. Els requisits de negoci existents i els fitxers de resultats associats a aquest projecte son esborrats.

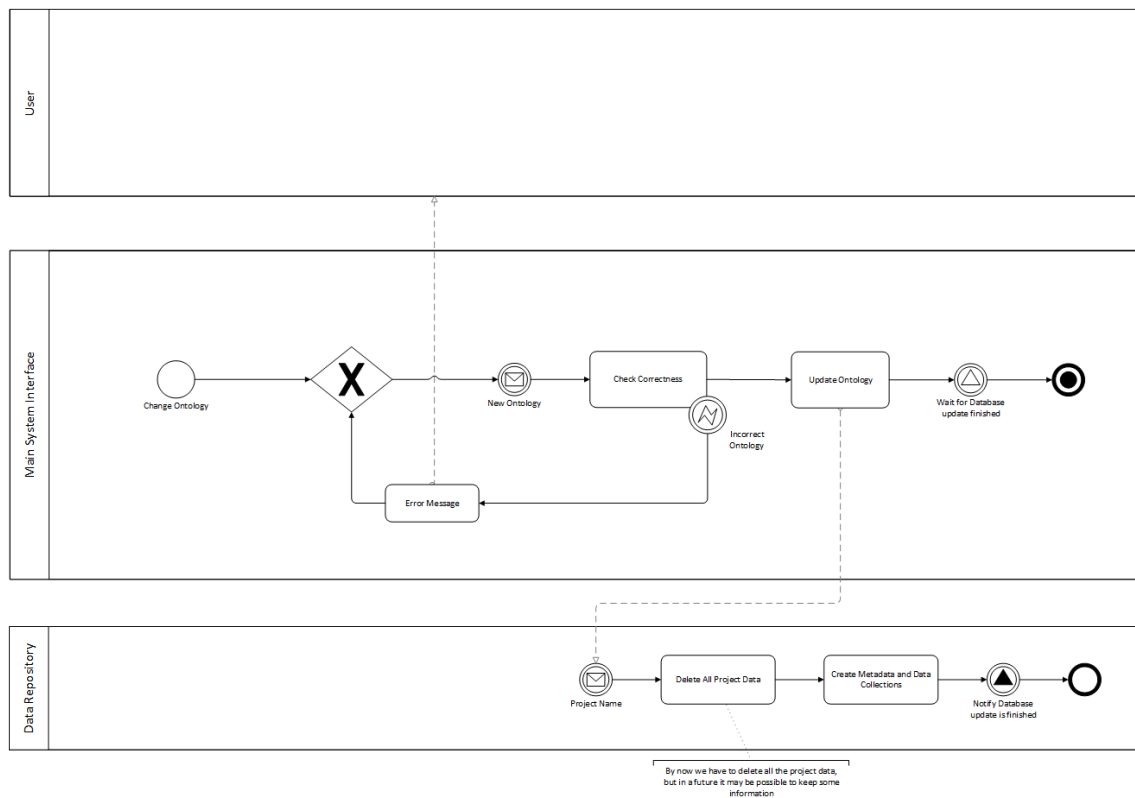


Figura 15: Diagrama BPMN del requisit #004 [Font: Propi]

### Requisit #005: Canviar el mapeig de les fonts

**Descripció:** L'usuari indica al sistema que vol canviar l'arxiu de mapeig de les fonts (*source mappings*) del projecte que es troba obert. El sistema adverteix a l'usuari que s'esborraran totes les dades associades al projecte exceptuant el nom del projecte (si en un futur cal preservar més dades ho indicarem). Seguidament l'usuari carrega la nova Ontologia i el sistema la guarda a la col·lecció de Metadades del repositori.

**Efecte Col·lateral:** Un cop el mapeig és canviat, tots els requisits de negoci i els fitxers de resultats del projecte han de ser invalidats per assegurar la consistència del sistema. Ara per ara el sistema esborrarà els fitxers del repositori de dades, però en futurs projectes es podrà dissenyar un mòdul per readaptar els requisits existents a la nova ontologia.

**Pre-condició:** Hem obert un projecte i l'usuari es troba a la pantalla de gestió de projectes.

**Post-condició:** El mapeig de les fonts associat al projecte obert és esborrat i un de nou és carregat. Els requisits de negoci existents i els fitxers de resultats associats a aquest projecte son esborrats.

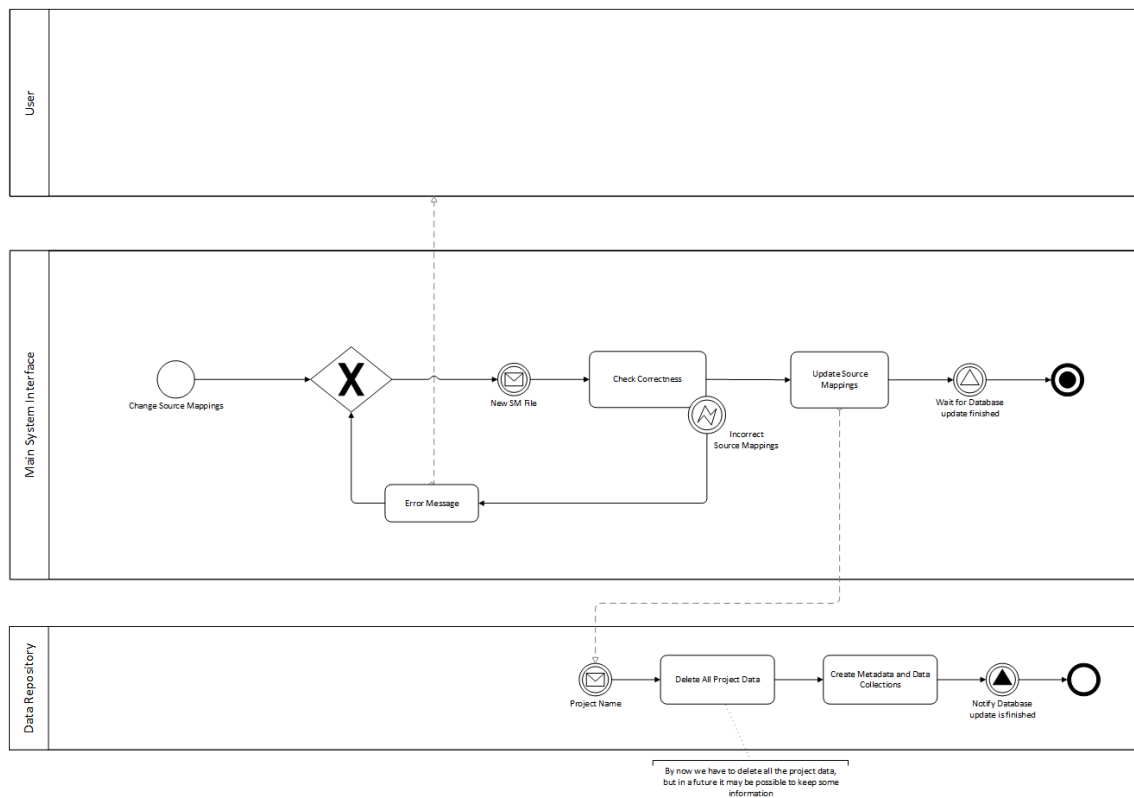


Figura 16: Diagrama BPMN del requisit #004 [Font: Propi]

### Requisit #006: Obrir el MineDigger

**Descripció:** Per tal de crear un nou requisit de negocis per a un projecte, l'usuari necessita obrir la eina MineDigger, el visor de grafs interactiu. L'usuari demana al sistema d'obrir el MineDigger i s'executa el requisit #009

El sistema demana, del repositori de dades, ho següent:

- El fitxer de la Ontologia
- El fitxer dels *Source Mappings*
- Requisits de negoci creats amb anterioritat.

Inicialment considerem que la ontologia i els mapeigs de les fonts es *parsejaren* de fitxer binari a XML de forma que el MineDigger els podrà entendre sense problema.

**Pre-condició:** Un projecte es troba obert i l'usuari es troba a la finestra de gestió del projecte.

**Post-condició:** El MineDigger es troba obert i llest per a crear requisits de negoci.

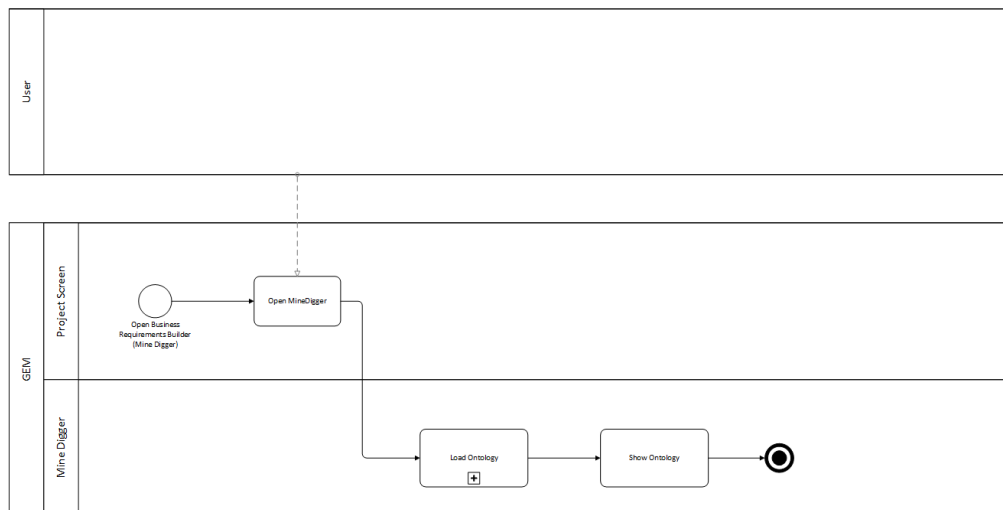


Figura 17: Diagrama BPMN del requisit #006 [Font: Propi]

### Requisit #007: Executar el procés principal (GEM, ORE i COAL)

**Descripció:** L'usuari indica que vol executar el procés principal, i el sistema mostra a l'usuari una llista de requisits de negoci dissenyats al projecte en el que l'usuari es troba. Un cop triat un requisit, el sistema executa el procés principal, passant a través de GEM, ORE i COAL i utilitzant el repositori de dades com a punt de contacte dels mòduls. El sistema genera un conjunt d'esquemes multi-dimensionals i processos ETL i els guarda a repositori.

Cal entendre que en aquest punt, aquests tres mòduls (GEM, ORE i COAL) es troben en desenvolupament. El nostre sistema estarà preparat per a executar el procés sencer, però algunes funcionalitats no es podran veure fins que els respectius projectes es completin (com ara el COAL, que es troba en una etapa encara inicial).

Seguidament el sistema mostra el visor de resultats amb els resultats que s'han obtingut del procés.

**Pre-condició:** Ens trobem a la pantalla de gestió de projectes i hi ha com a mínim un requisit de negoci definit al sistema.

**Post-condició:** El sistema ha generat un conjunt de resultats que es troben emmagatzemats a les col·leccions de grafs multidimensionals i ETLs del projecte al repositori. Ens trobem al visor de resultats, on es mostren aquest conjunt de resultats.

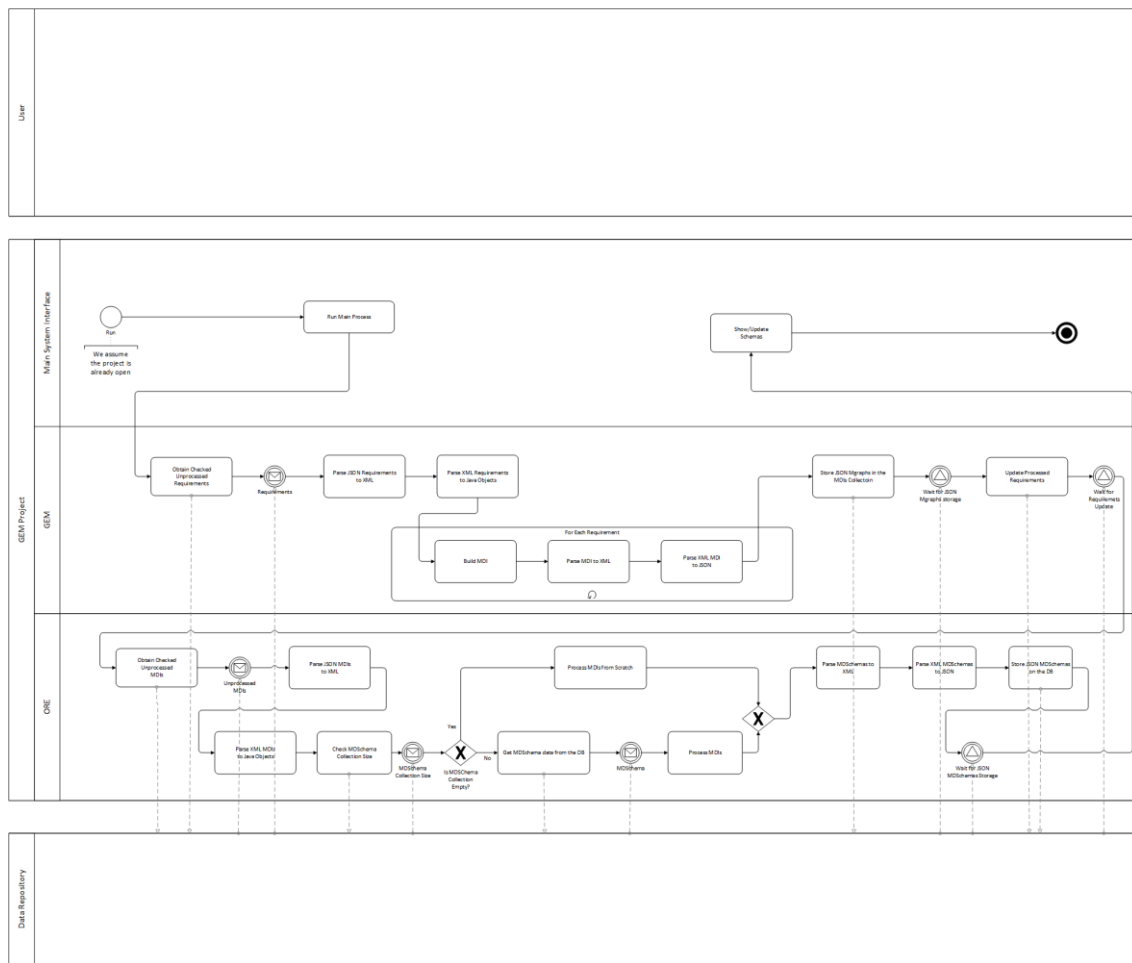


Figura 18: Diagrama BPMN del requisit #007 [Font: Propi]

### Requisit #008: Carregar els resultats del procés

**Descripció:** L'usuari indica al sistema que vol obrir resultats que s'han generat prèviament sobre algun requisit de negoci del projecte actual. El sistema mostra una llista desplegable mostrant els noms dels fitxers de resultat associats al projecte actual.

Seguidament l'usuari en tria un. El sistema recull del repositori de dades aquells elements (grafs multidimensionals i processos ETL) que componen aquest resultat. Aquests fitxers es troben en format JSON i descriuen la forma dels grafs i la informació necessària per a que la eina el pugui interpretar.

Finalment el sistema obre el visor de resultats i mostra el primer esquema del conjunt.

**Pre-condició:** Ens trobem a la pantalla de gestió de projectes i hi ha com a mínim un fitxer de resultats definit al sistema.

**Post-condició:** El visor de resultats es troba obert i el primer esquema del conjunt de resultats es mostra en pantalla.

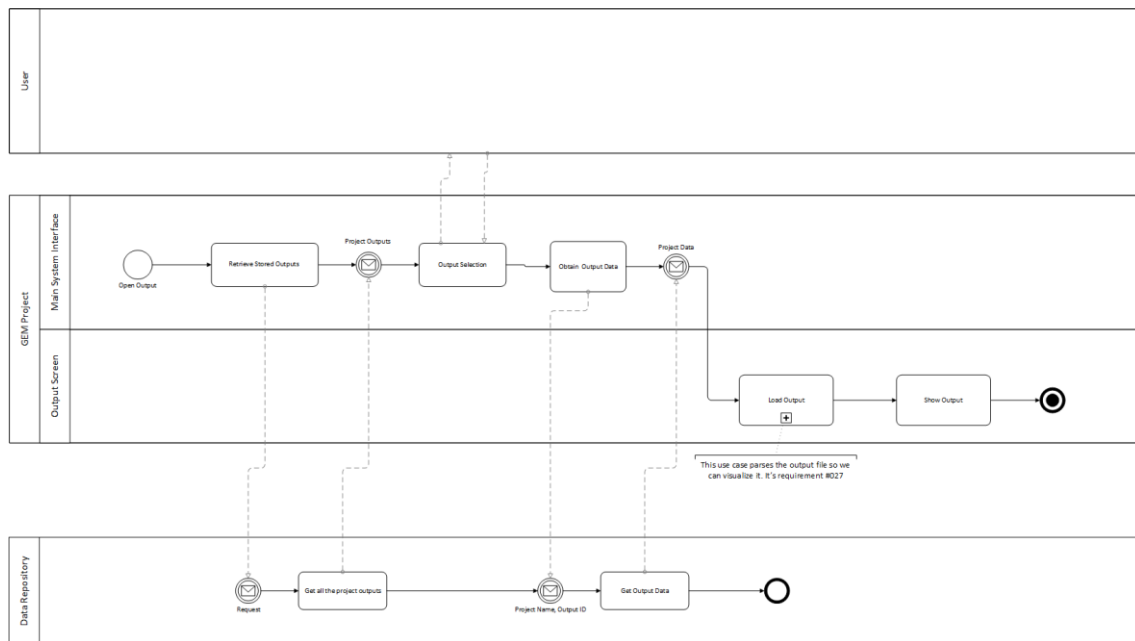


Figura 19: Diagrama BPMN del requisit #008 [Font: Propi]



Mapa navegacional inicial de la interfície principal del sistema

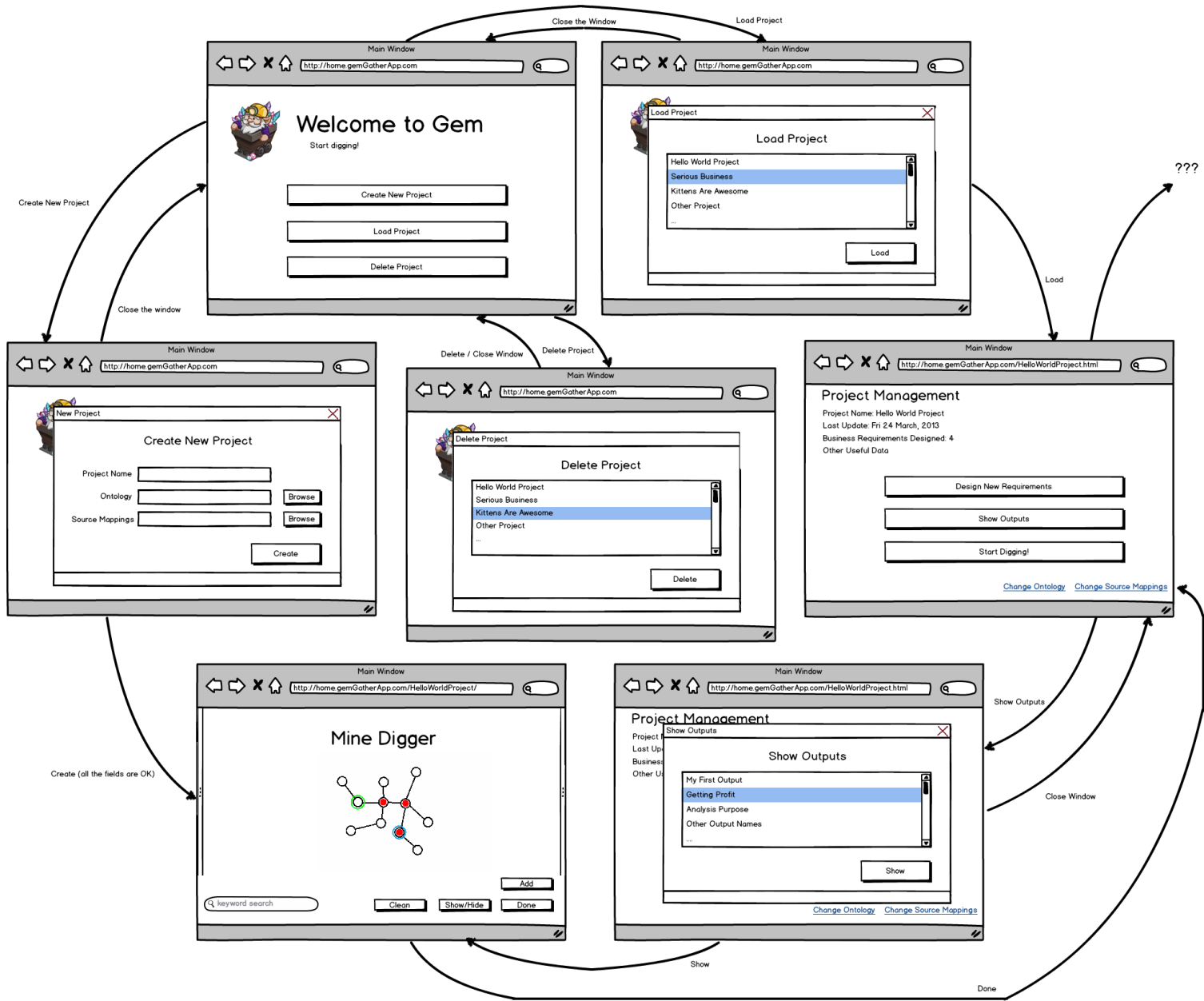


Figura 20: Mapa navegacional inicial del sistema [Font: Propi]

#### 6.2.2.2 MineDigger

##### **Requisit #009: Carregar arxius**

**Breu descripció:** Quan el MineDigger s'obre, el sistema necessita obtenir diversos fitxers de la base de dades interna. El sistema demana al repositori el següent:

- El fitxer de la Ontologia
- El fitxer dels *Source Mappings*
- Els requisits de negoci que s'hagin creat prèviament

La ontologia i els *Source Mappings* son parsejats de fitxer binari a XML, de forma que el MineDigger és capaç d'interpretar-los. Seguidament la ontologia és mostrada a la pantalla del MineDigger en forma de graf interactiu, i l'usuari pot començar a treballar sobre aquest graf per a muntar els seus requisits de negoci.

**Pre-condició:** L'usuari ha obert el MineDigger

**Post-condició:** El MineDigger es troba llest per a començar a dissenyar requisits de negoci sobre la ontologia associada al projecte.

##### **Requisit #010: Manipular el graf**

**Breu descripció:** El graf interactiu pot ser més gran que la pantalla on l'usuari es troba treballant, i algunes parts d'aquest podrien restar fora d'aquesta. Per tal de que l'usuari pugui treballar amb el graf sencer necessitem que el pugui moure.

Per a moure el graf l'usuari ha de fer un clic i mantenir amb el botó esquerre del ratolí sobre qualsevol node. Així, movent el ratolí, el graf es mourà amb ell fins que l'usuari deixi anar el botó del ratolí. No hem de confondre aquesta acció del ratolí (pitjar i mantenir) amb l'acció que es duu a terme al requisit #011 (pitjar i deixar anar).

**Pre-condició:** El MineDigger es troba obert.

**Post condició:** El graf s'ha mogut on l'usuari ha marcat.

##### **Requisit #011: Marcar un concepte com a rellevant**

**Breu descripció:** Quan es dissenya un requisit de negoci, l'usuari necessita marcar alguns conceptes com a rellevants per a ell. Tenim dues formes de marcar un concepte com a rellevant.

- Clic esquerre sobre un node no-rellevant
- Clic dret sobre un node no-rellevant. Apareixerà un menú desplegable del node triat. Aleshores fem clic sobre la opció "Rellevant".

Un cop el node és marcat com a rellevant, l'usuari obtindrà feedback visual (en aquest punt encara no és suficientment clar com serà aquest feedback. Ressaltar el node, donar-li un color de fons diferent, utilitzar etiquetes... tot depèn de la flexibilitat de l'eina).

**Pre-condició:** El MineDigger es troba obert. Com a mínim hi ha un node al graf que és no-rellevant.

**Post-condició:** El node seleccionat del graf s'ha marcat com a rellevant.

#### **Requisit #012: Marcar un concepte com a no-rellevant.**

**Breu descripció:** Pot succeir que l'usuari vulgui desfer l'acció de marcar un concepte com a rellevant, i que aquest torni a ser no-rellevant. De la mateixa manera que amb el requisit #011, l'usuari disposa de dues formes per tal d'assolir això:

- Clic esquerre sobre un node rellevant
- Clic dret sobre un node no-rellevant. Apareixerà un menú desplegable del node triat. Aleshores fem clic sobre la opció "No Rellevant".

Un cop el node s'ha marcat com a no-rellevant, l'usuari obtindrà feedback visual.

**Pre-condició:** El MineDigger es troba obert. Hi ha com a mínim un node al graf marcat com a rellevant.

**Post-condició:** El node seleccionat del graf s'ha marcat com a no rellevant.

#### **Requisit #013: Mostrar suggeriments de rol per als conceptes rellevants**

**Breu descripció:** Cada cop que l'usuari interactua amb el graf, poden aparèixer nous conceptes rellevants i relacionats, i conceptes que ja eren rellevants i relacionats poden deixar de ser-ho.

Aquest canvi implica un nou set de suggeriments de rol que el mòdul AMDO (Secció 2.2.1) pot donar a partir del nou estat de l'esquema. Aquest set de suggeriments encara no és prou clar si seran pre-computats al moment de carregar el projecte, o de forma dinàmica durant la interacció amb el graf, o per mitjà d'un *trigger* (botó o acció concreta).

Tampoc és prou clar si aquesta informació es guardarà al repositori, en una memòria cau o es perdrà un cop utilitzada. Durant els primers *sprints* s'anirà definint aquest comportament.

**Pre-condició:** El MineDigger es troba obert. L'usuari ha interaccionat amb el graf.

**Post-condició:** Si cal, nova informació apareix/s'amaga al graf, de forma que es compleix la següent declaració: Es mostren els suggeriments de rol multidimensional només per aquells conceptes que son rellevants o relacionats.

#### **Requisit #014: Definir el rol d'un concepte**

**Breu descripció:** A part dels suggeriments que dona AMDO, l'usuari pot triar explícitament el rol multidimensional d'un concepte. Per tal de fer-ho, l'usuari ha de fer clic dret sobre el concepte que vol definir i seguidament triar la opció "Definir com.." del menú desplegable que apareixerà.

Seguidament s'ofereixen les diferents opcions multidimensionals i l'usuari en tria una.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** El concepte triat ara té definit el rol multidimensional triat per l'usuari. Si hi havia algun suggeriment per part de l'AMDO, aquest s'amaga.

#### **Requisit #015: Tornar a mostrar suggeriments per a un concepte**

**Breu Descripció:** Si l'usuari ha triat manualment el rol d'un concepte (requisit #014), els suggeriments de l'AMDO s'han amagat per aquell concepte. Per tal de tornar a mostrar aquests suggeriments, l'usuari ha de fer clic dret al concepte i triar la opció "Definir com.." del menú desplegable que apareixerà. Seguidament cal triar "Detecció Automàtica" de les opcions que es mostren.

**Pre-condició:** El MineDigger es troba obert. L'usuari ha triat manualment el rol d'un concepte.

**Post-condició:** El concepte triat ara té definit el rol multidimensional suggerit per l'AMDO.

#### **Requisit #016: Amagar els conceptes no rellevants**

**Breu descripció:** Quan s'està construint un requisit de negoci pot resultar útil de veure aquelles parts del graf interactiu que son rellevants o relacionades i amagar les que no ho son. Així es mostra a l'usuari una imatge de quin és l'aspecte que té el requisit. Per tal de fer-ho, l'usuari ha de clicar un botó que es troba a la interfície.

**Pre-condició:** El MineDigger es troba obert. Els conceptes no rellevants i no relacionats no es troben amagats.

**Post-condició:** Els conceptes no rellevants i no relacionats s'amaguen.

#### **Requisit #017: Mostrar tots els conceptes**

**Breu Descripció:** Si el sistema està amagant els conceptes no rellevants i no relacionats (Requisit #016) l'usuari pot fer-los visibles de nou clicant un botó que es troba a la interfície.

**Pre-condició:** El MineDigger es troba obert. Els conceptes no rellevants i no relacionats es troben amagats.

**Post-condició:** Els conceptes no rellevants i no relacionats no es troben amagats.

#### **Requisit #018: Ressaltar una paraula clau**

**Breu descripció:** L'usuari escriu qualsevol paraula en un requadre de cerca. D'aquesta manera, qualsevol concepte del graf que coincideixi amb aquesta queda ressaltat.

Aquesta coincidència pot ser total o parcial. Tenim diferents opcions per definir què considerem com a coincidència:

- El sistema només ressaltarà aquells conceptes que coincideixin **totalment** amb la paraula o frase del requadre de cerca.
  - Requadre de cerca: *Preu del Menú* – Concepte: *Preu del Menú*
- El sistema només ressaltarà aquells conceptes que continguin com a mínim una **paraula** del requadre de cerca.
  - Requadre de cerca: *Preu* – Concepte: *Preu del Menú*
- El sistema només ressaltarà aquells conceptes que continguin el **text** del requadre de cerca.
  - Requadre de cerca: *Pr* – Concepte: *Preu del Menú*

**Possible Expansió:** Si només un concepte coincideix amb la cerca, aquest concepte se centrarà a la pantalla.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** Els conceptes que coincideixen amb el text del requadre de cerca es ressalten.

#### **Requisit #019: Esborrar tots els canvis del graf**

**Breu descripció:** L'usuari pot esborrar totes les modificacions que ha fet al graf interactiu i tornar a començar de zero. Per a fer-ho cal que pitgi el botó "Esborrar Tot" de la interfície. Apareixerà un missatge de confirmació.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** Totes les modificacions fetes al graf son esborrades. Això inclou els conceptes que hem marcat com a rellevants, els rols multidimensionals definits sobre els conceptes i les funcions d'agregació definides.

#### **Requisit #020: Comprovar la correctesa del requisit de negoci**

**Breu descripció:** Un requisit de negoci ha de seguir un conjunt de regles per a ser considerat com a correcte. No podem generar un requisit de negoci que es consideri invàlid.

Per a comprovar la correctesa d'un requisit de negoci disposem d'un fitxer de definicions DTD.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** El sistema sap si l'estat actual del graf es suficient per tal de construir un requisit de negoci. Si ho és, el botó de "Generar Requisit" es mostrarà activat. Si no, es mostrarà desactivat.

#### **Requisit #021: Generar un requisit de negoci**

**Breu descripció:** L'actual estat del graf interactiu serà parsejat com a requisit de negoci. Aquest *parsing* generarà un document JSON que resum totes les modificacions que l'usuari ha fet sobre el graf i serà emmagatzemat internament al repositori de dades, a la col·lecció de requisits associada al projecte actual.

**Pre-condició:** El MineDigger es troba obert. El l'estat del graf és correcte (Requisit #020).

**Post-condició:** S'ha generat un fitxer JSON que defineix el requisit de negoci i s'ha guardat al Minecart. Es mostra un missatge de confirmació.

#### **Requisit #022: Mostrar requisits prèviament dissenyats**

**Breu descripció:** Els requisits que s'han dissenyat prèviament es troben guardats internament al Minecart, a la col·lecció de requisits associada al projecte actual. L'usuari pot veure una llista amb aquests requisits en un *pane* lateral al MineDigger.

**Pre-condició:** El MineDigger es troba obert. Com a mínim s'ha dissenyat un requisit de negoci.

**Post-condició:** Els noms dels requisits de negoci dissenyats amb el sistema associats al projecte amb el que l'usuari està treballant apareixen llistats a un *pane* lateral.

#### **Requisit #023: Carregar requisits prèviament dissenyats**

**Breu descripció:** L'usuari pot veure quins requisits s'han dissenyat sobre el projecte en el que està treballant (Requisit #022) en un *pane* lateral. Clicant el botó corresponent el sistema ofereix la possibilitat de carregar un d'aquests requisits sobre el MineDigger. El sistema recupera el fitxer JSON que descriu el requisit del repositori de dades, a la col·lecció de requisits del projecte actual.

**Pre-condició:** El MineDigger es troba obert. S'executa el requisit #022

**Post-condició:** El graf interactiu descriu el requisit que l'usuari ha decidit carregar triat.

#### **Requisit #024: Esborrar requisits prèviament dissenyats**

**Breu descripció:** L'usuari pot veure quins requisits s'han dissenyat sobre el projecte en el que està treballant (Requisit #022) en un *pane* lateral. Clicant el botó corresponent el sistema ofereix la possibilitat d'esborrar un d'aquests requisits. El sistema esborra el fitxer JSON que descriu el requisit del repositori de dades, a la col·lecció de requisits del projecte actual.

**Pre-condició:** El MineDigger es troba obert. S'executa el requisit #022

**Post-condició:** Les dades relatives al requisit triat son esborrades del sistema. Qualsevol fitxer de resultats que hagués derivat d'utilitzar aquell requisit serà marcat com a inconsistent.

#### **Requisit #025: Canviar la paleta de colors**

**Breu descripció:** La majoria d'accions que es donen sobre el graf interactiu son representades per símbols o colors com a feedback visual. Podem canviar el color d'aquests visuals amb una paleta de color interactiva que es troba a la part inferior de la finestra.

D'aquesta manera, l'usuari pot personalitzar com es veu el graf.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** Tots els visuals que utilitzaven el color que l'usuari ha canviat, passen a ser del nou color triat.

#### **Requisit #026: Tancar l'editor**

**Breu descripció:** L'usuari pot tancar l'editor i tornar a la finestra de gestió de projecte. Per a fer-ho l'usuari ha de pitjar el botó "Endarrere" que es troba a la interfície. Si s'han fet modificacions des de l'últim cop que es guardar el requisit, un missatge d'alerta apareixerà informant a l'usuari que els canvis es perdran. Un cop l'usuari confirma el missatge l'editor es tanca.

**Pre-condició:** El MineDigger es troba obert.

**Post-condició:** El MineDigger es tanca. L'usuari es troba a la finestra de gestió de projecte del projecte actual.

#### **Mockups dels requisits**

Seguint la nostra metodologia, en aquest punt del projecte (Incepció) no podem proveir encara diagrames BPMN ni mapes navegacionals per al MineDigger, però podem mostrar un conjunt de *MockUps* que esbossen les idees principals de cada requisit i la seva visualització al navegador.

A la secció 8.3 trobem els mapes navegacionals reals del projecte una vegada acabada la seva implementació.

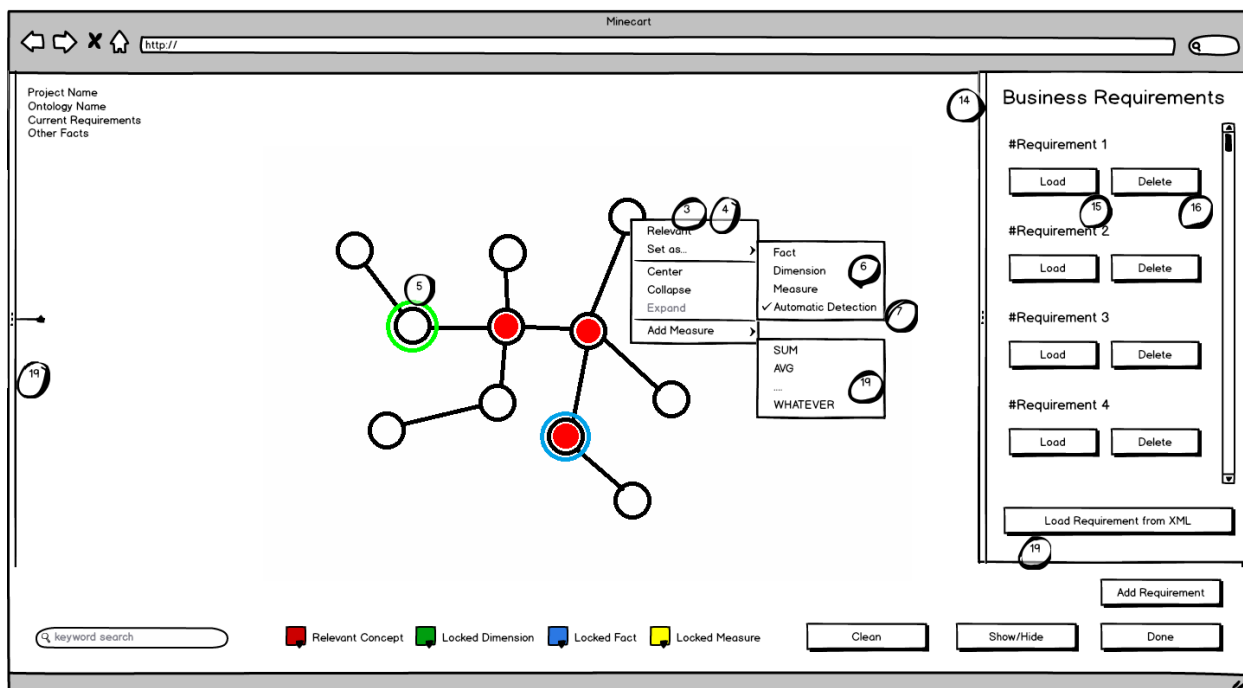
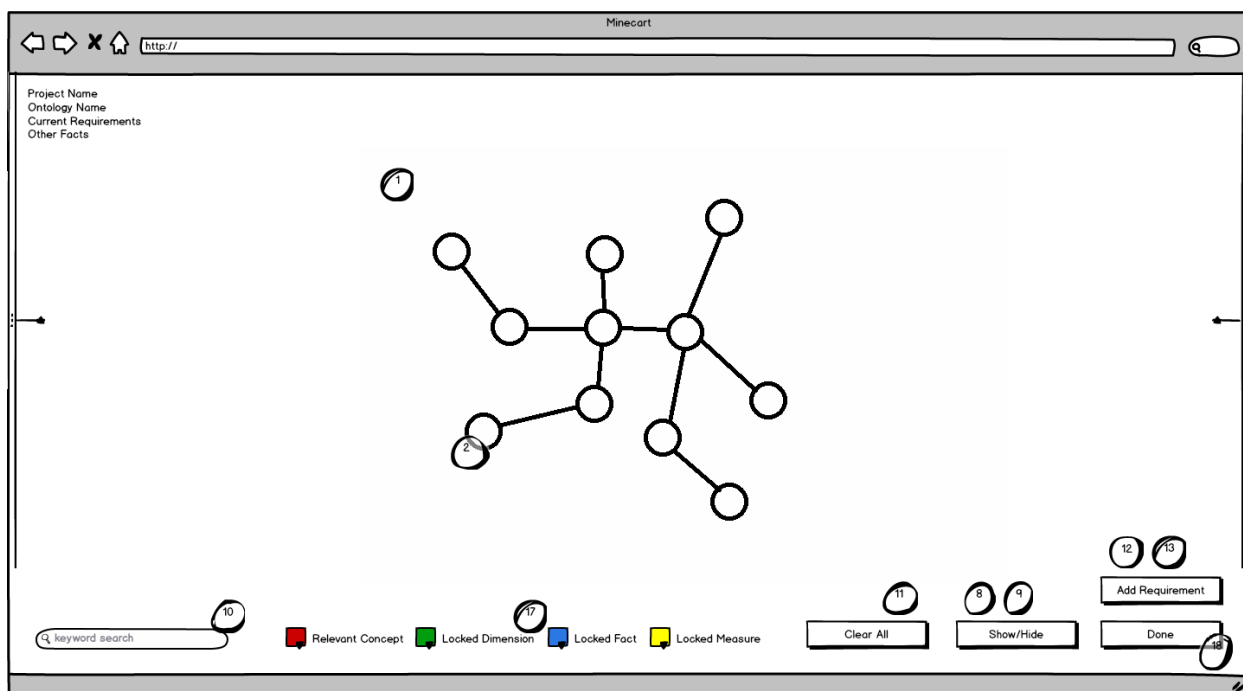


Figura 21: Mockup de les funcionalitats del MineDigger [Font: Propi]

- |   |  |
|---|--|
| 1- Carregar arxius  | 9- Mostrar tots els conceptes                      |
| 2- Manipular el graf  | 10- Ressaltar una paraula clau                     |
| 3- Marcar un concepte com a rellevant                                   | 11- Esborrar tots els canvis del graf              |
| 4- Marcar un concepte com a no rellevant                                | 12- Comprovar la correctesa del requisit de negoci |
| 5- Mostrar suggeriments de rol per a conceptes rellevants i relacionats | 13- Generar un requisit de negoci                  |
| 6- Definir el rol d'un concepte   | 14- Mostrar requisits prèviament dissenyats        |
| 7- Tornar a mostrar els suggeriments per a un concepte                  | 15- Carregar requisits prèviament dissenyats       |
| 8- Amagar els conceptes no rellevants i no relacionats                  | 16- Esborrar requisits prèviament dissenyats       |
|   | 17- Canviar la paleta de colors                    |
|   | 18- Tancar l'editor                                |
|   | 19- Tasques obertes                                |



### 6.2.2.3 Visor de resultats

#### **Requisit #027: Carregar fitxer de resultats**

**Breu descripció:** Per tal de poder visualitzar correctament els fitxers de resultats, hem de parsejar i tractar les dades que els componen. Així, al obrir un fitxer de resultats el sistema executarà una sèrie d'accions per tal d'obtenir fitxers JSON interpretables pel visor.

#### **Requisit #028: Canviar de diagrama**

**Breu descripció:** Un fitxer de resultats normalment conté més d'un diagrama. Grafs multidimensionals, processos ETL...

Utilitzant un sistema de pestanyes, llistes o botons podrem intercanviar els diagrames visibles.

#### **Requisit #029: Exportar el diagrama com una imatge**

**Breu descripció:** Tots els diagrames s'han de poder descarregar com a imatge (l'extensió encara està per determinar, depèn de l'eina).

#### **Requisit #030: Ressaltar una paraula clau**

**Breu descripció:** Quan l'usuari escriu qualsevol paraula en un requadre de cerca, qualsevol concepte del graf que coincideixi amb aquesta queda ressaltat.

#### **Requisit #031: Tancar el visor de resultats**

**Breu descripció:** L'usuari tanca el visor i el sistema el retorna a la pantalla de gestió de projecte.

### 5.2.2.4 Tasques Obertes

Al finalitzar aquest *sprint* hi ha tot un conjunt de requisits que encara no es troben definits del tot i per ara només son idees. També n'hi ha que depenen del desenvolupament d'altres mòduls, com el ORE, l'AMDO i el COAL. Aquests seran millor definits durant el desenvolupament del projecte. Son:

- Afegir accions per als nodes (expressions aritmètiques sobre les mesures, funcions, condicions i comparatives sobre descriptors de nivells, agregats sobre mesures...
- Carregar requisits usant fitxers XML i JSON creats fora del sistema.
- Cercar conceptes a la ontologia utilitzant una representació en arbre similar a la que s'utilitza al Protégé 3.5 [24]

## 6.3.Product Backlog Inicial

Derivat de l'especificació de requisits, dissenyem el *Product Backlog* del projecte per a començar a treballar.

Un cop acabat el projecte tornarem a mostrar el *Product Backlog* acabat, amb les modificacions i estimacions (Secció 7.13).

Identificació	Nom
US-001	Adaptar l'estructura del graf per tal de poder visualitzar els components de l'ontologia.
US-002	Crear un projecte nou
US-003	Obrir un projecte existent
US-004	Esborrar un projecte existent
US-005	Canviar la Ontologia del projecte
US-006	Canviar els Source Mappings del projecte
US-007	Obrir MineDigger
US-008	Fer funcionar el procés principal (GEM + ORE + COAL)
US-009	Carrega dels resultats del procés principal
US-010	Carregar Ontologies al MineDigger
US-011	Manipular el Graf
US-012	Marcar un concepte com a rellevant.
US-013	Marcar un concepte com a no rellevant
US-014	Definir el rol multidimensional d'un concepte
US-015	Mostrar el rol definit pels conceptes rellevants
US-016	Amagar els conceptes no-rellevants i no-relacionats
US-017	Mostrar tots els conceptes
US-018	Ressaltar una paraula clau del MineDigger
US-019	Esborrar totes les modificacions fetes al graf
US-020	Comprovar la correctesa del requisit de negoci
US-021	Generar un requisit de negoci a partir del graf
US-022	Mostrar requisits prèviament generats.
US-023	Carregar requisits prèviament generats.
US-024	Esborrar Requisits prèviament generats.
US-025	Canviar la Paleta de Color dels nodes
US-026	Tancar l'editor
US-027	Afegir accions sobre els nodes
US-028	Mostrar/amagar suggeriments de rol pels conceptes rellevants
US-029	Explorar els conceptes de l'Ontologia amb una representació de tipus arbre
US-030	Carregar resultats
US-031	Canviar entre els diferents diagrames al visor de resultats
US-032	Exportar el graf multidimensional com una imatge
US-033	Exportar el procés ETL com un fitxer Kettle
US-034	Exportar el esquema multidimensional com un conjunt de sentències DDL SQL
US-035	Ressaltar una paraula clau de visor de resultats
US-036	Tancar la pantalla de resultats

## 7. Implementació

En aquesta secció trobem tota la documentació relacionada amb la Fase 2 del projecte, que consisteix en la implementació del projecte. Durant aquesta fase seguint amb la nostra metodologia, dividim el temps en períodes anomenats *sprints* i intentem assaltar les diferents *user stories* definides al product backlog.

A continuació adjuntem la documentació dels *sprints*. Cada *sprint* consta de les següents parts:

- Llista de les *user stories* que s'han volgut assaltar aquella setmana, especificant el criteri d'acceptació de cadascuna, i indicant si s'han superat o no, i per quins motius. També es mostra el temps que es preveia que es trigaria i el temps real dedicat a cadascuna. Cal tenir en compte que les estimacions temporals tenen en compte no només el **temps d'implementació**, sinó també tot l'**aprenentatge** que s'hi troba implícit.
- Llista (si n'hi ha) de les *user stories* que s'han modificat i/o eliminat, i per quins motius.
- Llista (si n'hi ha) de les *user stories* que han aparegut arrel de la discussió de la reunió de l'*sprint*.
- Informació addicional relativa a la reunió o al projecte que pot haver aparegut, tal com esquemes, diagrames o imatges del progrés o d'algunes funcionalitats.

### 7.1 Sprint 1

Data de la reunió: 25/03/2014

**Resum de l'*sprint*:** Durant aquest primer *sprint* ens hem centrat en tractar les tecnologies que es volen utilitzar i comprovar si compleixen els requisits adequats. Per a la visualització gràfica s'ha decidit, d'entre un ventall de possibilitats, d'utilitzar una llibreria gràfica en JavaScript anomenada d3.js [25].

Arrel de diversos exemples que s'ofereixen al lloc web [26] s'ha intentat validar si aquesta eina serà suficient per als propòsits del projecte. També s'ha començat a treballar amb l'eina del servidor, on s'estan estudiant diverses possibilitats tals com utilitzar *frameworks* (Play! [27] i node.js [28]) o un Apache Tomcat [29], utilitzar JSP, servlets o fer córrer JavaScript directament dins una classe Java.

#### **User Stories tractades:**

- US-037: La tria de l'eina de visualització gràfica és adequada
  - Criteri d'acceptació:
    - És possible canviar l'aspecte visual del graf: Tamany, color, links i formes.
    - És possible ressaltar alguns nodes amb diferents colors.
    - Els nodes reaccionen a esdeveniments del ratolí (clic dret, clic esquerre...).
    - És possible moure el graf per l'escena.
    - És possible contraure i expandir les diferents branques del graf.
    - Es pot construir un graf a partir d'un fitxer JSON ben formatat.
    - Es poden crear nodes nous en temps d'execució.
    - Es poden destruir nodes en temps d'execució.
    - Es poden posar etiquetes de text sobre els nodes.

- Es poden amagar/mostrar diferents nodes en temps d'execució.
- Es poden mostrar finestres modals quan es clica sobre un node.
- Es poden mostrar finestres modals quan es passa el ratolí sobre un node.
- Resultat: Acceptat
- Temps estimat per la tasca: 10h
- Temps real: 20h

#### User Stories noves:

- US-038: S'han de poder guardar els canvis i l'estat del graf en un document JSON
  - Motivació: Hem pogut comprovar que d3.js treballa amb uns documents que tenen unes característiques i una estructura molt concreta. L'usuari ha de poder guardar la seva feina i tornar a recuperar-la en qualsevol moment, de forma que s'ha de poder guardar l'estat del graf de forma que l'eina sigui capaç d'interpretar-ho, i restaurar-ho.
- US-039: La tria de l'eina per al servidor és adequada.
  - Motivació: Hi ha molts tipus d'eines diferents al mercat per a poder treballar amb servidors. Nosaltres necessitem una arquitectura on es pugui treballar amb Java a la banda del servidor i JavaScript a la banda del client, i ha de resultar-nos senzilla i pràctica (ja que és el primer cop que treballem amb aquest tipus d'eina).

## 7.2 Sprint 2

Data de la reunió: 01/04/2014

**Resum de l'sprint:** Durant aquest *sprint* s'ha seguit treballant amb l'eina. Ens hem centrat en intentar representar un graf propi amb d3.js utilitzant una pàgina JSON. Fins al moment aquesta pàgina JSON està generada a ma, però caldrà dissenyar un pàrser que ens transformi ontologies .owl en fitxers JSON suficients per al nostre projecte.

#### User Stories tractades:

- US-010: Carregar ontologies al MineDigger
  - Criteri d'acceptació:
    - Donat un fitxer amb extensió .json que representa una Ontologia, aquesta es mostra en pantalla en forma de graf.
    - El fitxer JSON cal que es trobi formatat correctament.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 5h
  - Temps real: 10h
- US-011: Manipular el Graf
  - Criteri d'acceptació:
    - Fent una acció (*drag and drop* a un dels nodes) amb el ratolí, podem moure els nodes del graf.
    - Un cop hem mogut el graf els nodes tornen a establitzar-se de nou.

- Movent els nodes podem aconseguir donar formes diferents al graf en cas que no ens agradi com estan disposats.
- Resultat: Acceptat
- Temps estimat per la tasca: 2h
- Temps real: 7h

#### **User Stories noves:**

- US-040: La visualització de l'ontologia és adequada i clara per a l'usuari.
  - Motivació: Al carregar el nostre graf, l'eina ens mostra els nodes i les connexions d'una forma amb la que resulta poc intuïtiu de treballar per a l'usuari (com mostrem a la figura 22).

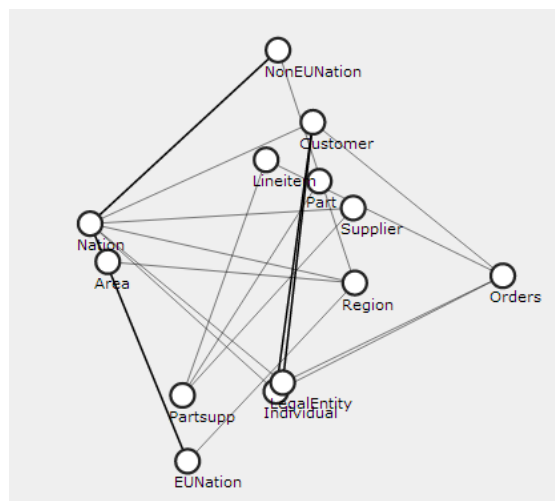


Figura 22: Exemple de parametrització errònia i visualització incorrecta del graf [Font: Propi]

Així doncs, ens cal definir una funció que permeti visualitzar aquest graf correctament.

- US-041: Podem carregar qualsevol ontologia que tingui format .owl directament a la nostra eina.
  - Motivació: Els usuaris han de poder carregar les seves pròpies ontologies al sistema. Ens cal un conjunt d'eines o pàrsers que siguin capaces d'obrir aquest tipus de documents i transformar-los en un JSON formatat adequadament.
- US-042: Disposem d'una base de dades interna NoSQL capaç de manejar totes les dades del sistema.
  - Motivació: Amb l'ús del software s'hauran de guardar diversos tipus de dades. Metadades, dades temporals, fitxers de resultats... Ens cal dissenyar una solució que serveixi per a l'emmagatzemament d'aquestes dades.
- US-043: La base de dades ha de ser accessible des del Servidor
  - Motivació: El servidor ha de poder accedir a la base de dades per a l'intercanvi de dades entre els diferents mòduls (AMDO, GEM, ORE i COAL).

- US-044: La base de dades ha de ser accessible des del Client
  - Motivació: El client ha de poder fer crides a la base de dades per tal de guardar/recuperar la feina, crear projectes nous, carregar ontologies...

## 7.3 Sprint 3

Data de la reunió: 08/04/2014

**Resum de l'sprint:** Seguim treballant amb l'eina de visualització, intentant preparar el terreny per a les funcionalitats que ens fan falta. A la vegada s'està realitzant un *training* intens en JavaScript, ja que l'estudiant mai ha tocat aquest llenguatge. Poc a poc s'adverteix el funcionament i es va generant l'estructura del programa.

Per a treballar amb ontologies s'està utilitzant Jena, una llibreria en Java que, entre d'altres funcionalitats, pot treballar amb fitxers .owl.

Ahora s'està treballant amb MongoDB per a muntar la base de dades interna.

### **User Stories tractades:**

- US-041: Podem carregar qualsevol ontologia que tingui format .owl directament a la nostra eina.
  - Criteri d'acceptació:
    - Disposem d'una eina que al rebre qualsevol fitxer .owl ens retorna un fitxer JSON amb totes les dades que identifiquen la Ontologia i son necessàries pel programa.
    - El fitxer JSON cal que es trobi formatat correctament.
    - Quan el MineDigger rep aquest fitxer el pot mostrar com a graf.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real: 10h
- US-042: Disposem d'una base de dades interna NoSQL capaç de manejar totes les dades del sistema.
  - Criteri d'acceptació:
    - El disseny de la base de dades és correcte i ens permet guardar totes les dades necessàries.
    - El disseny aprofita les avantatges de la tecnologia NoSQL
    - Podem realitzar connexions a la base de dades i dur a terme les operacions bàsiques: insercions, modificacions i esborrats.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 6h
  - Temps real: 7h

- Adjuntem un diagrama de l'estructura d'aquesta base de dades:

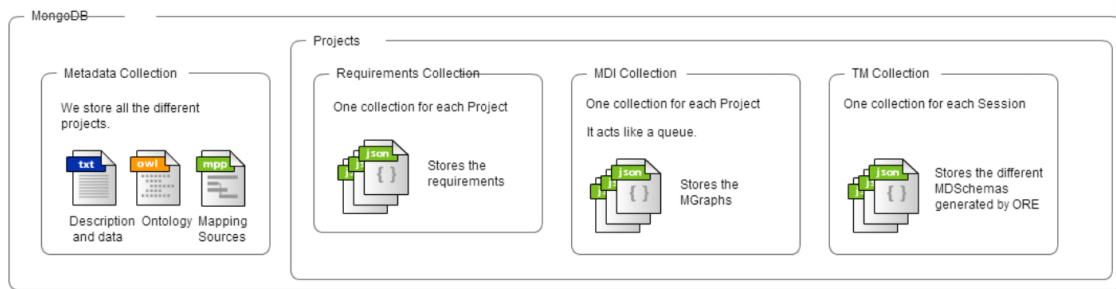


Figura 23: Primera proposta d'estructura de la base de dades interna [Font: Propi]

#### User Stories noves:

- US-045: Cada concepte té associat una etiqueta que en mostra el nom.
  - Motivació: Al adonar-nos que l'eina no suporta directament adjuntar texts als nodes, cal dissenyar una funció que se n'encarregui.
- US-046: Apareixen requadres contextuais sobre els nodes al passar el ratolí per sobre.
  - Motivació: Hem arribat a la conclusió de que pot ser bo per a l'usuari conèixer els *datatypes* associats a cada concepte de la ontologia. Així, caldria fer aparèixer un requadre contextual quan l'usuari passa el ratolí per sobre mostrant el nom i el tipus de dada dels *datatypes* associats a cada concepte.

## 7.4 Sprint 4

Data de la reunió: 15/04/2014

**Resum de l'sprint:** Durant aquest *sprint* hem assaltat algunes de les funcionalitats bàsiques del programa. Ara que ja podem carregar grafs és el moment de resoldre una per una totes les tasques que cal fer.

Actualment som capaços de mostrar ontologies de forma que l'usuari pot veure'n tots els conceptes de forma simple. També pot realitzar accions simples com marcar i desmarcar els nodes.

Cal remarcar que les tasques que hem assaltat en aquest *sprint* han sigut més complicades del que pensàvem en un principi, i com es pot veure el temps de treball real difereix en gran mesura de la planificació. Resulta difícil conciliar totes les llibreries que estem utilitzant en un sol document JavaScript.

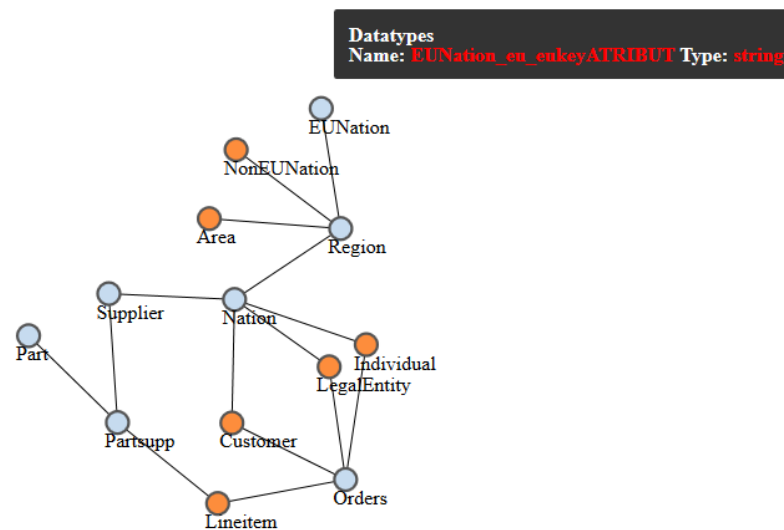


Figura 24: Exemple de la visualització de la ontologia d'exemple TPC-H, amb el requadre contextual que en mostra els atributs (US-046) [Font: Propi]

#### User Stories tractades:

- US-001: Adaptar l'estructura del graf per tal de poder visualitzar els components de la ontologia.
  - Criteri d'acceptació:
    - Cada node del graf representa una classe de la ontologia.
    - Els atributs associats a les classes de la ontologia es troben definits internament al node del graf.
    - Les propietats que defineixen relacions entre conceptes de la ontologia son representades com a links entre els diferents nodes.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real: 1h
- US-012: Marcar un concepte com a rellevant.
  - Criteri d'acceptació:
    - Sempre que l'usuari faci una acció de clic esquerre sobre un dels nodes que representi un concepte no rellevant, aquest canviarà el seu estat a rellevant.
    - Sempre que l'usuari faci una acció de clic dret sobre un dels nodes que representi un concepte no rellevant, es desplegarà un menú contextual. Entre les múltiples opcions que presentarà aquest menú, trobem una anomenada *Set as Non Relevant*. Si l'usuari fa clic esquerre sobre aquesta opció, el node canviarà el seu estat a rellevant.
    - Un node que contingui un concepte rellevant tindrà el contorn pintat d'un color que serà diferent als nodes que continguin un concepte no rellevant.
    - Guardarem el nou estat d'aquest concepte com a atribut del node.



- Resultat: Acceptat
- Temps estimat per la tasca: 5h
- Temps real: 10h
- US-013: Marcar un concepte com a no-rellevant.
  - Criteri d'acceptació:
    - Sempre que l'usuari faci una acció de clic esquerre sobre un dels nodes que representi un concepte rellevant, aquest canviarà el seu estat a no rellevant.
    - Sempre que l'usuari faci una acció de clic dret sobre un dels nodes que representi un concepte rellevant, es desplegarà un menú contextual. Entre les múltiples opcions que presentarà aquest menú, trobem una anomenada *Set as Relevant*. Si l'usuari fa clic esquerre sobre aquesta opció, el node canviarà el seu estat a rellevant.
    - Un node que contingui un concepte no rellevant tindrà el contorn pintat d'un color que serà diferent als nodes que continguin un concepte rellevant.
    - Guardarem el nou estat d'aquest concepte com a atribut del node.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 0.5h
- US-040: La visualització de l'ontologia és adequada i clara per l'usuari.
  - Criteri d'acceptació:
    - La parametrització de la funció *force* de la llibreria d3 és tal que els nodes compleixen les característiques següents:
      - Els nodes es troben prou separats entre si.
      - Les etiquetes de text son llegibles.
      - Els links entre nodes s'entrecreuen el mínim possible
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-045: Cada concepte té associat una etiqueta que en mostra el nom.
  - Criteri d'acceptació:
    - Per cada node que forma el graf, trobem una etiqueta de text que en mostra el nom.
    - El nom que es mostra és el nom de la classe de la ontologia.
    - La etiqueta es mostra sempre en una posició de (-10px, -10px) en relació al centre del node.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 2h

- US-046: Apareixen requadres contextuais sobre els nodes al passar el ratolí per sobre.
  - Criteri d'acceptació:
    - Quan el ratolí es troba sobre l'àrea del node un requadre flotant apareix sobre el node.
    - Aquest requadre conté tots els atributs que representen *datatypes* associats amb la classe de la ontologia representada pel node.
    - Per cada *datatype* se'n mostra el nom i el tipus de dada (int, float, string..)
    - Quan el ratolí surt de l'àrea del node el requadre desapareix.
    - El requadre sempre es mostra per sobre de qualsevol altre element SVG.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 13h

## 7.5 Sprint 5

Data de la reunió: 22/04/2014

**Resum de l'sprint:** Un dels aspectes més complicats dins la nostra aplicació és el de mostrar a l'usuari totes les accions que pot fer sobre els nodes d'una forma simple i no invasiva. El primer en que hem pensat és en utilitzar un menú desplegable que s'obre quan fem clic dret sobre un dels nodes. És una tasca que vam començar a realitzar a l'sprint anterior, però que cal continuar perfilant.

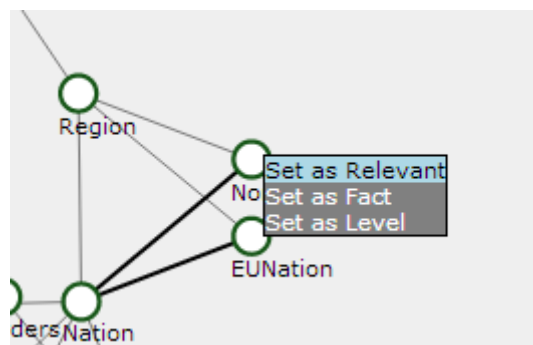


Figura 25: Menú desplegable que apareix al fer clic dret sobre el node [Font: Propi]

Tot i semblar una tasca simple, al estar executant l'aplicació sobre el navegador ens ha resultat bastant complicat, ja que normalment els esdeveniments de clic dret son ja interpretats per aquest. Tot i així hem trobat la forma de realitzar-ho.

### User Stories tractades:

- US-014: Definir el rol multidimensional d'un concepte.
  - Criteri d'acceptació:
    - Sempre que l'usuari faci una acció de clic dret sobre un dels nodes, es desplegarà un menú contextual. Entre les múltiples opcions que presentarà aquest menú, trobem una anomenada *Set as Fact* i una anomenada *Set as Level*. Si l'usuari fa clic esquerre sobre aquesta opció, el node canviarà el seu rol multidimensional a fet o a dimensió, segons hagi triat l'usuari.
    - Guardarem el nou estat d'aquest concepte com a atribut del node.

- Resultat: Acceptat
- Temps estimat per la tasca: 7h
- Temps real: 10h
- US-015: Mostrar el rol definit pels conceptes rellevants.
  - Criteri d'acceptació:
    - Per cada concepte rellevant, pintarem el node del graf que el representa d'un color concret.
    - Aquest color serà diferent segons si és un concepte que hem definit manualment com a fet, si l'hem definit manualment com a dimensió o si volem deixar que l'AMDO sigui el que decideixi el rol.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real: 3h

#### **User Stories noves:**

- US-047: Marcar els descriptors rellevants d'un nivell
  - Motivació: L'usuari ha de poder marcar quins descriptors son rellevants per al requisit de negoci, de forma que GEM els pugui tenir en compte.
- US-048: Definir restriccions sobre els descriptors d'un nivell
  - Motivació: L'usuari ha de poder definir restriccions sobre els descriptors de forma que GEM les pugui tenir en compte.
- US-049: Definir diferents focus d'anàlisi sobre un fet
  - Motivació: L'usuari ha de poder definir diferents focus d'anàlisi sobre els descriptors de forma que GEM els pugui tenir en compte.

#### **User Stories ampliades**

- US-027: Afegir accions sobre els nodes
  - Motivació: Hem ampliat aquesta *user story* per que resulta massa complexa com per a tractar-la per individual. L'hem descomposada en les *user stories* US-047, US-048 i US-049.

## **7.5 Sprint 6**

Data de la reunió: 06/05/2014

**Resum de l'sprint:** Aquest *sprint* suposa un punt d'inflexió dins el treball. Després d'estar l'estudiant dues setmanes incapacitat per malaltia, s'ha hagut de dur a terme un treball de replanificació força fort per a poder quadrar tots els requisits a l'entrega final (Secció 8.1 i Secció 8.2)

Tot i així, es preveu que es podrà continuar sense problemes amb el projecte a les dades acordades.

La durada d'aquest *sprint*, per les causes nomenades, ha sigut de dues setmanes. S'han vist, però nombrosos avenços al projecte:

En primer lloc s'ha substituït el menú d'accions de clic dret per una finestra modal una mica més elegant i més manegable per a l'usuari (Figura 26).

Utilitzem una llibreria anomenada tip.d3[26] amb la que, mitjançant la creació dinàmica d'HTML i l'ús de JavaScript podem mostrar les diferents opcions que es poden fer sobre els nodes.

Podem veure que ara es poden realitzar certes accions, com marcar els descriptors i atributs rellevants. De qualsevol forma aquesta interfície encara es troba en procés de desenvolupament i s'han d'afegir moltes accions.

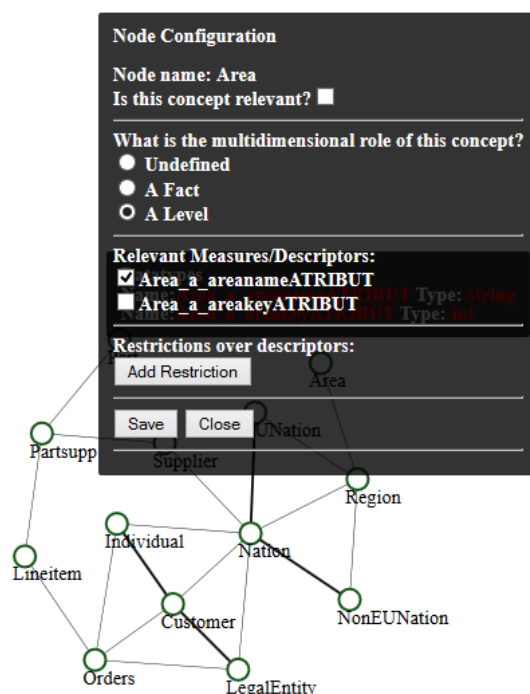


Figura 26: Finestra contextual de configuració dels nodes [Font: Propi]

Ens hem adonat que cal donar molta importància a aquests petits factors que en un principi consideràvem irrelevants. Apareixen (i apareixeran) noves *User Stories* relacionades amb aquest menú.

#### **User Stories tractades:**

- US-047: Marcar els descriptors rellevants d'un nivell.
  - Criteri d'acceptació:
    - Al fer clic sobre un node se'ns mostra una finestra sobre aquest. És el panell de configuració del node.
    - El rol multidimensional del node es mostra seleccionat entre les diferents opcions.

- Si el rol multidimensional del concepte és el de *Level* es mostren, entre altres coses, la llista d'atributs o *datatypes* associats amb aquest. Aquests seran els possibles descriptors.
- Cada descriptor ve acompanyat d'una *checkbox*.
- L'usuari pot marcar o desmarcar les *checkboxes* com desitgi.
- Si l'usuari canvia el rol multidimensional del concepte, es guarda temporalment quins son els descriptors que l'usuari ha marcat. Així, si l'usuari torna a definir el rol multidimensional del concepte com a *Level*, es tornen a marcar.
- Si l'usuari tria la opció de *Save*, es guarda dins la informació del node la llista de descriptors rellevants.
- Si l'usuari torna a obrir el panell de configuració del node, es recupera la llista de descriptors rellevants i es marquen les *checkboxes* pertinents.
- Si l'usuari ha canviat el rol multidimensional del concepte i tria la opció *Save* es perd tota la informació relativa als descriptors.
- Resultat: Acceptat
- Temps estimat per la tasca: 15h
- Temps real: 20h

#### **User Stories modificades:**

- US-014: Definir el rol multidimensional d'un concepte.
  - Motivació: Amb la creació d'aquesta nova finestra de configuració del node, cal readaptar com definim el rol multidimensional del concepte. Eliminarem el menú de clic dret i readaptarem la finestra per a poder definir el rol allà mateix. Resulta més útil centralitzar totes les opcions sota una mateixa finestra.
  - Criteri d'acceptació:
    - Al fer clic sobre un node se'ns mostra una finestra sobre aquest. És el panell de configuració del node.
    - Podem veure les diferents opcions de rol multidimensional que es poden definir sobre el node: *Level*, *Fact* i *Undefined*.
    - La opció predefinida serà la de *Undefined*.
    - Al costat de cada opció hi ha un *RadioButton*. L'usuari pot triar així el rol desitjat. Cada cop que canvia el rol, canvien les opcions i l'aspecte del panell de configuració del node.
    - Si l'usuari tria la opció de *Save*, es guarda dins la informació del node el seu rol.
    - Si l'usuari torna a obrir el panell de configuració del node, es trobarà marcat el seu rol i es mostraran les opcions pertinents.
  - Resultat: Modificat i Acceptat

#### **User Stories ampliades:**

- US-025: Canviar la paleta de color dels nodes
  - Motivació: La nova estructura del sistema ens fa que necessitem dividir aquesta *user story* en tres: US-051, US-054 i US-055, amb una granularitat més fina.

#### **User Stories noves:**

- US-050: Mostrar la llista d'atributs en una *box* quan aquests son massa llargs
  - Motivació: Ens hem adonat que la llista de mesures dels nodes pot arribar a resultar molt llarga y això és un problema, ja que la finestra contextual pot resultar massa gran. Així, hem decidit que cal mostrar les mesures en una *boxlist* amb barra de desplaçament.
- US-051: Mostrar una llegenda
  - Motivació: Cal una correspondència entre els colors del graf i el seu significat, de forma que l'usuari sempre pugui conèixer el context del que veu.
- US-052: Es mostra de forma visual la jerarquia de classes al graf
  - Motivació: Ara per ara el graf mostra les relacions conceptuais entre els nodes, però en una ontologia podem trobar relacions jeràrquiques. Aquesta informació pot resultar d'utilitat per a contextualitzar l'usuari. A mes a mes, en un futur serà interessant plantejar funcionalitats que juguin amb aquesta jerarquia, com per exemple el col·lapsament i expansió de branques jeràrquiques senceres.
- US-053: Exportar els fitxers ETL com a imatges
  - Motivació: Considerem que serà útil poder exportar no només els grafs multidimensionals com a imatges, sinó també els processos ETL.
- US-054: Llegenda interactiva per al MineDigger
  - Motivació: La llegenda mostra a l'usuari la correspondència entre color i concepte, però també pot ser interessant deixar a l'usuari triar quin color correspon amb què, de forma que farem la llegenda interactiva per a l'usuari. Així, quan aquest exporti les imatges, tindran l'aspecte que ell desitgi.
- US-055: Llegenda interactiva per al visor de resultats
  - Motivació: Igual que amb la US-054, serà interessant disposar d'una llegenda interactiva per al visor de resultats també.
- US-056: Exportar els requisits com a imatges
  - Motivació: Considerem que serà útil poder exportar no només els grafs multidimensionals com a imatges, sinó també els dissenys dels requisits.

## **7.7 Sprint 7**

Data: 13/05/2014

**Resum de l'sprint:** Durant aquest *sprint* hem enfocat el nostre treball en acabar la finestra de configuració dels nodes i començar a muntar el servidor. Hem decidit muntar un Apache Tomcat i dissenyar diversos *servlets* que seran els que ens permetran comunicar la banda del client amb la del servidor, tal i com mostra la figura 27.

Hem considerat que utilitzar servlets era molt millor opció que utilitzar Java Server Pages (JSP), ja que els processos que es duren a terme al servidor requereixen de moltes classes i tenen una estructura massa complexa per a representar-los amb una sola JSP.

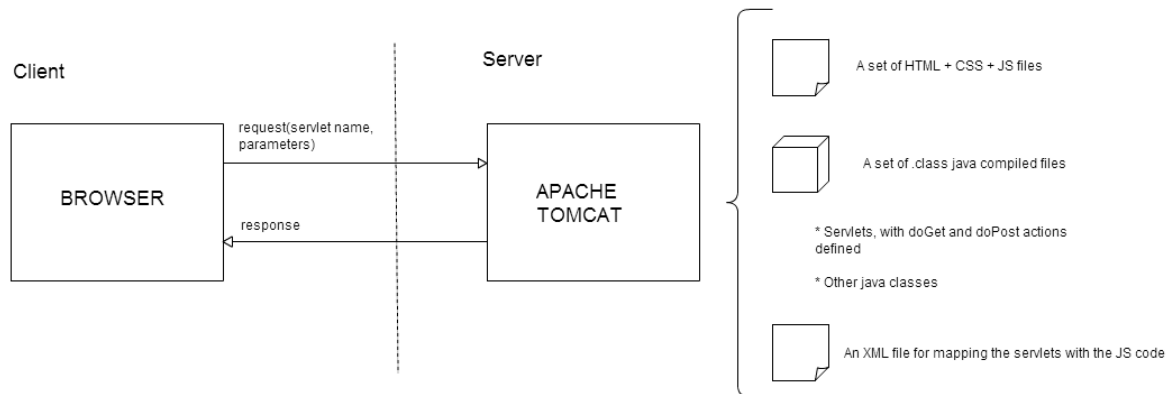


Figura 27: Esquema de la distribució client-servidor de la aplicació [Font: Propi]

Hem dissenyat i implementat les diferents opcions de la pantalla de configuració de nodes de la següent manera:

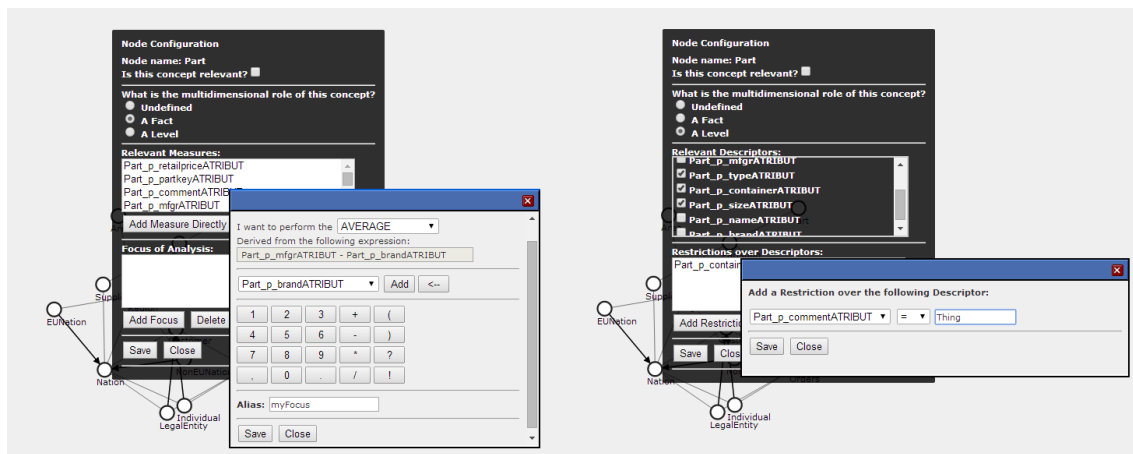


Figura 28: Finestra modal de configuració dels nodes [Font: Propi]

A la figura podem veure com canvia la pantalla de configuració depenent de com definim el node, i quin aspecte tenen les finestres modals de definició d'*slicers* (restriccions sobre mesures) i focus d'anàlisi (US-048 i US-049).

#### User Stories tractades:

- US-039: La tria de l'eina per al servidor és adequada
  - Criteri d'acceptació:
    - Podem executar codi Java.
    - Podem definir respostes a crides GET i POST.
    - Podem adjuntar paràmetres a les crides GET i POST, així com codi JSON.
    - Podem enviar codi JSON com a resposta a les crides GET i POST.

- Es pot treballar amb *threads*.
- Resultat: Acceptat
- Temps estimat per la tasca: 5h
- Temps real: 7h
- US-043: La base de dades ha de ser accessible des del Servidor.
  - Criteri d'acceptació:
    - Disposem d'una API que ens permet de realitzar accions sobre la base de dades des de les classes Java.
    - Aquestes accions engloben la inserció, esborrat, modificat i cerca de dades sobre la base de dades.
    - També cal suportar projeccions i condicions sobre aquestes accions.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 3h
- US-048: Definir restriccions sobre els descriptors d'un nivell
  - Criteri d'acceptació:
    - Al fer clic sobre un node se'ns mostra una finestra sobre aquest. És el panell de configuració del node.
    - El rol multidimensional del node es mostra seleccionat entre les diferents opcions.
    - Si el rol multidimensional del concepte és el de *Level*, es mostren, entre altres coses, la llista d'atributs o *datatypes* i una *listbox* que mostra les restriccions sobre els descriptors, també anomenades *slicers*.
    - L'usuari pot definir noves restriccions sobre els descriptors. Al fer clic sobre el botó *Add new Restriction* es mostra una finestra modal que permet definir aquest *slicer* (Figura 28)
    - Si l'usuari canvia el rol multidimensional del concepte, es guarden temporalment les restriccions definides. Així, si l'usuari torna a definir el rol multidimensional del concepte com a *Level*, aquestes tornen a aparèixer.
    - Si l'usuari tria la opció *Save*, es guarda dins la informació del node la llista restriccions definides.
    - Si l'usuari torna a obrir el panell de configuració del node, es recupera la llista de descriptors rellevants i es marquen les *checkboxes* pertinents.
    - Si l'usuari ha canviat el rol multidimensional del concepte i tria la opció *Save* es perd tota la informació relativa a les restriccions.
    - L'usuari pot esborrar els diferents *slicers* seleccionant-los a la *listbox* i pitjant el botó *Delete*.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 10h
  - Temps real: 10h



- US-049: Definir diferents focus d'anàlisi sobre un fet
  - Criteri d'acceptació:
    - Al fer clic sobre un node se'ns mostra una finestra sobre aquest. És el panell de configuració del node.
    - El rol multidimensional del node es mostra seleccionat entre les diferents opcions.
    - Si el rol multidimensional del concepte és el de *Fact*, es mostren, entre altres coses, la llista d'atributs o *mesures* i una *listbox* que mostra els diferents focus d'anàlisi.
    - L'usuari pot definir nous focus d'anàlisi fent clic sobre el botó *Add new focus of analysis*. Es mostrarà una finestra modal que permet definir aquest focus d'anàlisi (Figura 28) L'usuari també pot definir un nou focus d'anàlisi a partir d'una mesura. Per a fer-ho cal que el seleccioni de la *listbox* i pitgi el botó *Add Measure Directly*.
    - Si l'usuari canvia el rol multidimensional del concepte, es guarden temporalment els focus d'anàlisi definits. Així, si l'usuari torna a definir el rol multidimensional del concepte com a *Fact*, aquestes tornen a aparèixer.
    - Si l'usuari tria la opció *Save*, es guarda dins la informació del node la llista restriccions definides.
    - Si l'usuari torna a obrir el panell de configuració del node, es recupera la llista de descriptors rellevants i es marquen les *checkboxes* pertinents.
    - Si l'usuari ha canviat el rol multidimensional del concepte i tria la opció *Save* es perd tota la informació relativa als focus d'anàlisi.
    - L'usuari pot editar els diferents focus d'anàlisi seleccionant-los a la *listbox* i pitjant el botó *Edit*.
    - L'usuari pot esborrar els diferents focus d'anàlisi seleccionant-los a la *listbox* i pitjant el botó *Delete*.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 10h
  - Temps real: 13h
  -

#### **User Stories Posposades**

- US-018: Ressaltar una paraula clau del MineDigger
  - Motivació: La dificultat per integrar un cercador de paraules clau amb les llibreries gràfiques.
- US-035: Ressaltar una paraula clau del visor de resultats
  - Motivació: La mateixa motivació que amb la US-018.

## 7.8 Sprint 8

Data de la reunió: 20/05/2014

**Resum de l'sprint:** Les funcionalitats del MineDigger comencen a consolidar-se. Durant aquest *sprint* hem volgut centrar-nos en la interfície del sistema. Així, hem començat a dissenyar les pantalles per a fer les diferents accions: Crear, obrir, esborrar i administrar projectes.

Hem hagut de crear els primer servlets del sistema, i comprovar que el servidor funciona sense problemes. A la secció 9.3 podem veure el mapa navegacional complet del sistema, amb totes aquestes pantalles.

### **User Stories tractades:**

- US-002: Crear un projecte nou
  - Criteri d'acceptació:
    - La pantalla principal del sistema mostra la opció de crear un projecte nou.
    - Al seleccionar aquesta opció, es presenta un formulari. L'usuari ha d'emplenar el formulari amb el nom del projecte i els fitxers de l'ontologia i els *Source Mappings*
    - Al pitjar el botó *New Project* es fa una crida POST al *servlet* adequat.
    - El projecte queda guardat al repositori de dades. Els fitxers associats s'emmagatzemen a la col·lecció de *bindata* associada al projecte.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 4h
- US-003: Obrir un projecte existent
  - Criteri d'acceptació:
    - La pantalla principal del sistema mostra la opció d'obrir un projecte existent.
    - Al seleccionar aquesta opció, es presenta una llista amb els projectes existents. Aquesta informació l'obtenim de la col·lecció de fitxers del projecte al repositori de dades utilitzant un *servlet* i una crida POST.
    - Quan l'usuari tria un projecte i pitja la opció *Open Project* s'obre la finestra de gestió del projecte.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-004: Esborrar un projecte existent
  - Criteri d'acceptació:
    - La pantalla principal del sistema mostra la opció d'esborrar un projecte existent.
    - Al seleccionar aquesta opció, es presenta una llista amb els projectes existents. Aquesta informació l'obtenim de la col·lecció de metadades del Minecart utilitzant un *servlet* i una crida POST.
    - Quan l'usuari tria un projecte i pitja la opció *Delete Project*, s'elimina el contingut de les col·leccions de requisits, *MGraphs* (grafs

multidimensionals) i ETLs i fitxers de la base de dades utilitzant un *servlet*, així com el projecte de la col·lecció de metadades del repositori de dades.

- Resultat: Acceptat
- Temps estimat per la tasca: 1h
- Temps real: 1h
- US-007: Obrir el MineDigger
  - Criteri d'acceptació:
    - La pantalla de gestió de projecte mostra les opcions de dissenyar nous requisits i editar els requisits ja existents.
    - Ambdues opcions obren el MineDigger. Utilitzem una crida POST amb les dades necessàries. Es carrega la ontologia (US-001) utilitzant el fitxer de la col·lecció de fitxers del projecte.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-026: Tancar l'editor
  - Criteri d'acceptació:
    - Sempre que l'usuari pitgi el botó *Close* del Mine Digger, es tancarà l'editor i es tornarà a la finestra de gestió de projectes.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 0.5h
- US-038: S'han de poder guardar els canvis i l'estat del graf en un document JSON
  - Criteri d'acceptació:
    - Un pàrser interpreta i transforma el graf i totes les accions que l'usuari ha realitzat en un document JSON.
    - El document conté 2 interpretacions. La primera divideix el contingut del graf en 4 seccions: Agregats, Dimensions, Mesures i *Slicers*. Aquesta interpretació serà la que utilitzarà GEM.
    - La segona interpretació descriu cada node amb totes les accions que l'usuari ha realitzat.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 5h

## 7.9 Sprint 9

Data de la reunió: 27/05/2014

**Resum de l'sprint:** El principal objectiu d'aquest *sprint* és el de aconseguir que el procés principal funcioni. Caldrà adaptar GEM per a que llegeixi els requisits del repositori intern de dades, i aconseguir que els resultats es guardin també a la base de dades.

A la vegada treballarem altres funcionalitats com l'emmagatzemament de requisits, amagar i mostrar conceptes...

També hem canviat la forma en la que guardem i carreguem els requisits. Hem decidit que es farà a la pantalla de gestió de projectes, enlloc de fer-ho en un pane lateral, ja que utilitzarem el pane lateral per a altres afers.

#### **User Stories tractades:**

- US-005: Canviar la Ontologia del projecte
  - Criteri d'acceptació:
    - A la finestra de gestió de projecte hi ha un botó per a canviar els fitxers del projecte.
    - Al clicar el botó, s'adverteix a l'usuari que s'eliminaran els requisits i fitxers de resultats associats amb el projecte, ja que resultaran invàlids.
    - Seguidament es mostra una pantalla per tal de carregar la nova ontologia i el nou fitxer de mapeig de dades (US-006).
    - Al confirmar, s'esborren els requisits i els fitxers de resultats associats al projecte actual (col·leccions de Requisits, *MGraphs* (grafs multidimensionals) i ETL del projecte) i es pugen a la col·lecció de metadades els fitxers que ha triat l'usuari.
    - Al acabar, es torna a la finestra de gestió de projecte.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 0.5h
- US-006: Canviar els *Source Mappings* del projecte
  - Criteri d'acceptació:
    - A la finestra de gestió de projecte hi ha un botó per a canviar els fitxers del projecte.
    - Al clicar el botó, s'adverteix a l'usuari que s'eliminaran els requisits i fitxers de resultats associats amb el projecte, ja que resultaran invàlids.
    - Seguidament es mostra una pantalla per tal de carregar la nova ontologia (US-005) i el nou fitxer de mapeig de dades.
    - Al confirmar, s'esborren els requisits i els fitxers de resultats associats al projecte actual (col·leccions de Requisits, *MGraphs* (grafs multidimensionals) i ETL del projecte) i es pugen a la col·lecció de metadades els fitxers que ha triat l'usuari.
    - Al acabar, es torna a la finestra de gestió de projecte.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 0.5h
- US-008: Fer funcionar el procés principal (GEM+ORE+COAL)
  - Criteri d'acceptació:
    - A la finestra de gestió de projecte ha un botó (*Start Digging!*) per a fer funcionar el procés principal.

- Al clicar el botó, es mostren a l'usuari els requisits emmagatzemats al sistema relacionats amb el projecte actual.
    - L'usuari pot triar un nom per al fitxer de resultats que es generarà. El nom per defecte serà el del requisit que ha triat l'usuari.
    - Si el nom que ha triat l'usuari ja existeix a la base de dades per aquell projecte, es notificarà a l'usuari de que ha de canviar-lo.
    - Un cop l'usuari tria requisit, un nom, i pitja el botó de *Dig*, es crida al *servlet* que s'encarrega de fer anar el GEM.
    - GEM genera un seguit de resultats que s'emmagatzemen al repositori de dades, a les col·leccions de *MGraphs* (grafs multidimensionals) i ETLs corresponents.
    - Al acabar, s'obre el visor de resultats i es mostra el primer graf resultant del procés.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 10h
  - Temps real: 15h
- US-016: Amagar els conceptes no rellevants i no relacionats
    - Criteri d'acceptació:
      - Al MineDigger hi ha un botó per amagar i mostrar els conceptes no rellevants i no relacionats (*Hide/Show relevant concepts*).
      - Sempre que els nodes no estiguin ja amagats, si l'usuari pitja aquest botó, els nodes que representen conceptes no rellevants i no relacionats desapareixen. Tots els links que tenen un dels dos extrems en un d'aquests nodes desapareixen també. Les etiquetes de text associades al node desapareixen. Els nodes només desapareixen visualment, ja que la informació associada a aquests es manté.
      - Els nodes que no s'estan mostrant tampoc son receptius a interaccions.
      - El botó canvia la seva etiqueta de text, passant a ser ara *Show relevant concepts*.
    - Resultat: Acceptat
    - Temps estimat per la tasca: 2h
    - Temps real: 2h
  - US-017: Mostrar tots els conceptes
    - Criteri d'acceptació:
      - Al MineDigger hi ha un botó per a amagar i mostrar els conceptes no rellevants i no relacionats (*Hide/Show relevant concepts*).
      - Sempre que els nodes que representen conceptes no rellevants i no relacionats estiguin amagats, si l'usuari pitja aquest botó els nodes es mostraran. Tots els links que tenen un dels dos extrems en un d'aquests nodes es mostraran també. Les etiquetes de text associades al node es mostren.
      - El botó canvia la seva etiqueta de text, passant a ser ara *Hide relevant concepts*.

- Resultat: Acceptat
- Temps estimat per la tasca: 1h
- Temps real: 1h
  
- US-021: Generar un requisit de negoci a partir del graf
  - Criteri d'acceptació:
    - L'usuari pot donar un nom al requisit de negoci que està generant en una etiqueta de text.
    - Quan l'usuari pitja el botó *Save Requirement* es llença el pàrser que transforma el graf en dos documents JSON.
    - Aquests documents es guarden al repositori de dades a la col·lecció de requisits del projecte.
    - Es notifica a l'usuari que la operació s'ha realitzat amb èxit.
    - Si existeix un requisit amb el nom triat, es demana a l'usuari si el vol sobreesciure o cancel·lar la operació.
    - Si l'usuari no ha especificat un nom per al requisit, el sistema mostra un missatge indicant que cal donar un nom.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 4h
  - Temps real: 3h
  
- US-022: Mostrar requisits prèviament generats
  - Criteri d'acceptació:
    - La pantalla principal del sistema mostra la opció d'obrir un requisit existent.
    - Al seleccionar aquesta opció, es presenta una llista amb els requisits existents relacionats amb el projecte actual.
    - Els requisits es mostren en una *listbox*, i s'obtenen de la col·lecció de requisits del projecte utilitzant un *servlet* i una petició POST.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
  
- US-023: Carregar requisits prèviament generats
  - Criteri d'acceptació:
    - L'usuari pot triar un requisit de la llista de requisits generats.
    - Un cop triat, pot pitjar el botó *Edit Requirements* per a carregar aquest requisit al MineDigger. El graf interactiu tindrà el mateix aspecte que quan es va guardar el requisit.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 0.5h
  
- US-024: Esborrar requisits prèviament generats
  - Criteri d'acceptació:
    - L'usuari pot triar un requisit de la llista de requisits generats.

- Un cop triat, pot pitjar el botó *Delete* per a esborrar aquest requisit de la base de dades. Utilitzarem un *servlet* i una petició POST.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 0.5h
- US-044: La base de dades ha de ser accessible des del Client
  - Criteri d'acceptació:
    - Disposem d'un conjunt de *servlets* que ens permeten de realitzar les següents accions:
      - Obtenir un element de la base de dades de qualsevol col·lecció.
      - Definir projeccions i condicions com a paràmetres de cerca.
      - Emmagatzemar un element o conjunt d'elements a la base de dades a qualsevol col·lecció.
      - Esborrar un element o conjunt d'elements de la base de dades de qualsevol col·lecció.
      - Esborrar una col·lecció de la base de dades.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 4h

## 7.10 Sprint 10

Data de la reunió: 03/06/2014

**Resum de l'*sprint*:** Ja s'acosta la data final, de forma que aquest *sprint* hem volgut finalitzar el major nombre de funcionalitats del MineDigger. Podem veure com als primers *sprints* el nombre d'*user stories* tractades era menor, però comportaven un major treball i complexitat, i ara podem tractar molts més *sprints* en un temps menor. Això es deu a que el grau actual d'expertesa amb les tecnologies és superior i que el treball associat amb les *User Stories* és menor (funcionalitats secundàries).

### **User Stories tractades:**

- US-009: Càrrega dels resultats del procés principal
  - Criteri d'acceptació:
    - La pantalla de gestió de projectes mostra la opció d'obrir un fitxer de resultats.
    - Al seleccionar aquesta opció, es presenta una llista amb els fitxers de resultats existents al sistema relacionats amb el projecte actual en una *listbox*, i s'obtenen de la col·lecció de *MGraphs* (grafs multidimensionals) del projecte utilitzant un *servlet* i una petició POST.
    - Quan l'usuari tria un dels fitxers i pitja el botó *Show Outputs*, s'obre el visor de resultats i es mostra el primer *MGraph* del conjunt de resultats del fitxer en forma de graf interactiu.

- Resultat: Acceptat
- Temps estimat per la tasca: 1h
- Temps real: 0.5h
  
- US-019: Esborrar totes les modificacions fetes al graf
  - Criteri d'acceptació:
    - Al MineDigger trobem una botó que l'usuari pot utilitzar per a poder esborrar totes les modificacions i canvis que l'usuari hagi pogut fer al graf.
    - Quan l'usuari tria aquesta opció, l'estat del graf interactiu canviarà, esborrant-se tots els *slicers*, focus d'anàlisi, descriptors, i passant tots els nodes a ser conceptes no rellevants.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real: 2h
  
- US-020: Comprovar la correctesa del requisit de negoci
  - Criteri d'acceptació:
    - Un fitxer DTD ens marca les pautes que fan que un requisit de negoci sigui correcte i estigui ben estructurat.
    - El sistema sempre sabrà si el requisit de negoci que s'està construint és correcte.
    - S'ofereix a l'usuari la possibilitat de saber si el requisit que està muntant és correcte o no pitjant un botó. El sistema mostrarà una finestra modal indicant-ho.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
  
- US-030: Carregar resultats
  - Criteri d'acceptació:
    - Els fitxers de resultats han d'estar correctament parsejats per tal de que el d3.js els pugui interpretar correctament.
    - Per les representacions multidimensionals marcarem quins son els nodes i de quin tipus (informació dimensional o informació sobre els conceptes, nodes principals o atributs) i com estan connectats entre si. Representarem els atributs amb nodes més petits i acurtarem la distància entre aquests i el node que acompanyen.
    - Per les representacions de processos ETL utilitzarem un seguit de classes que ens generen un mapeig per coordenades i mostrarem el flux com a graf estàtic utilitzant aquestes coordenades. Afegirem detalls sobre el contingut de cada node en una finestra modal que podem veure si fem clic sobre aquest.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 4h
  - Temps real: 5h



- US-032: Exportar el graf multidimensional com una imatge
  - Criteri d'acceptació:
    - El visor de resultats conté un botó que ens permet exportar l'aspecte visual del graf multidimensional com una imatge.
    - Aquesta imatge té format .svg, un format que s'utilitza per a descriure components gràfics HTML.
    - La imatge es podrà obrir amb qualsevol navegador modern.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 2h
  
- US-036: Tancar la pantalla de resultats
  - Criteri d'acceptació:
    - Sempre que l'usuari pitgi el botó *Close* del visor de resultats, es tancarà l'editor i es tornarà a la finestra de gestió de projectes.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 0.5h
  
- US-050: Mostrar la llista d'atributs en una *box* quan aquests son massa llargs
  - Criteri d'acceptació:
    - La finestra de configuració dels nodes conté una secció que mostra els diferents descriptors de la classe, quan aquesta és un *Level*.
    - Aquesta llista es mostra en una *listbox* de tamany 5, *scrollable*, i acompanyats dels *checkboxes* pertinents.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real 1h
  
- US-052: Es mostra de forma visual la jerarquia de classes al graf
  - Criteri d'acceptació:
    - Les classes que tenen entre si una relació jeràrquica es trobaran *linkades*. Aquest link serà de tipus *link* i tindrà un *marker-end* en forma de punta de fletxa a la classe pare.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 3h
  
- US-053: Exportar els fitxers ETL com a imatges
  - Criteri d'acceptació:
    - El visor de resultats conté un botó que ens permet exportar l'aspecte visual del procés ETL com una imatge.
    - Aquesta imatge té format .svg, un format que s'utilitza per a descriure components gràfics HTML.

- La imatge es podrà obrir amb qualsevol navegador modern.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 0.5h
- US-054: Llegenda interactiva per al MineDigger
  - Criteri d'acceptació:
    - A la part inferior del MineDigger hi ha una llegenda de colors que marca de quin color és cada tipus de node: Nodes rellevants, nodes no rellevants, conceptes, fets i nivells.
    - Quan l'usuari fa clic sobre un dels colors es desplega una paleta de colors i l'usuari pot triar el color que desitgi.
    - Automàticament tots els nodes d'aquest tipus canvien el seu color al nou color triat.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 4h
  - Temps real: 8h
- US-055: Llegenda interactiva per al visor de resultats
  - Criteri d'acceptació:
    - A la part inferior del visor de resultats hi ha una llegenda de colors que marca de quin color és cada tipus de node: Conceptes, mesures, dades de tipus *groupBy*, i altres tipus de dades dimensionals.
    - Quan l'usuari fa clic sobre un dels colors es desplega una paleta de colors i l'usuari pot triar el color que desitgi.
    - Automàticament tots els nodes d'aquest tipus canvien el seu color al nou color triat.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-056: Exportar els requisits de negoci com a imatges
  - Criteri d'acceptació:
    - El MineDigger conté un botó que ens permet exportar l'aspecte visual de requisit com una imatge.
    - Aquesta imatge té format .svg, un format que s'utilitza per a descriure components gràfics HTML.
    - La imatge es podrà obrir amb qualsevol navegador modern.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 0.5h
  - Temps real: 0.5h

#### **User Stories noves:**

- US-057: El graf interactiu pot anclar els nodes o be simular forces
  - Motivació: Hem trobat durant els moments de test que de vegades pot resultar incòmode la simulació de forces del graf. Volem afegir una opció per a fixar els nodes i deixar que l'usuari els mogui sense que cap força hi influeixi.
- US-058: Es mostren missatges de confirmació quan s'acaben els processos principals
  - Motivació: Hi ha processos que no son instantanis, i l'usuari no te forma de saber quan s'han acabat i quan pot continuar utilitzant el software. Necessitem mostrar missatges quan aquests processos finalitzen.
- US-059: La finestra de que mostra els atributs dels nodes desapareix al fer *drag*
  - Motivació: Hem trobat durant els moments de test que la finestra contextual que mostra els atributs dels nodes no desapareix mentre estem movent-lo.
- US-060: Millora del *Look and Feel* del sistema
  - Motivació: Estem utilitzant HTML pur, sense cap tipus de millora visual, ja que fins ara estem treballant a un nivell funcional. Ara que comencem a tenir el sistema quasi finalitzat, ens cal una millora en el *look and feel*, i fer que aquest sigui atractiu a la vista.
- US-061: Canvi dinàmic de l'estat dels botons d'accions
  - Motivació: Hi ha accions i botons que no sempre es podran utilitzar, per exemple la d'obrir un projecte quan no s'ha seleccionat cap projecte. Cal que els botons restin inactius quan no es puguin fer servir.
- US-062: Les accions del MineDigger es mostren en un pane lateral desplegable
  - Motivació: Resulta més còmode tenir les accions agrupades en un *pane* lateral que com a botons enmig de l'àrea de treball. A mes a mes guanyarem espai.
- US-063: Esborrar fitxers de resultats
  - Motivació: Un cop l'usuari ha generat fitxers de resultats pot ser necessari esborrar-los del sistema.

## **7.11 Sprint 11**

Data de la reunió: 10/06/2014

**Resum de l'sprint:** Un cop acabat el MineDigger, hem centrat aquest *sprint* a acabar les funcionalitats del visor de resultats. Carregar fitxers, canviar de diagrames, i algunes altres funcionalitats menors relacionades amb la interfície.

També hem començat a plantejar les tasques de la Fase 3, a nivell de recopilar i generar la documentació final.

### **User Stories tractades:**

- US-031: Canviar entre els diferents diagrames al visor de resultats
  - Criteri d'acceptació:
    - Al visor de resultats disposem de dos botons: *Next* i *Previous*.
    - Clicant aquests botons podrem navegar a través de les diferents interpretacions multidimensionals del fitxer de resultats.
    - Disposem també d'un botó anomenat: *Show ETL*.
    - Clicant aquest botó podrem veure el procés ETL associat al graf multidimensional actiu.
    - Els botons *Next* i *Previous* es desactivaran quan ens trobem al primer i a l'últim diagrama respectivament.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-057: El graf interactiu pot anclar els nodes o be simular forces
  - Criteri d'acceptació:
    - El comportament per defecte d'un graf de la llibreria d3 és el de simular forces que reaccionen als *clics* i als *drags*.
    - El graf també pot mantenir els seus nodes estàtics (permetent, això si, moviments de *click-and-drag*).
    - El MineDigger disposa d'un botó que permet a l'usuari canviar aquests modes entre si. La etiqueta de text del boto canviarà entre *Fix* i *Unfix*
  - Resultat: Acceptat
  - Temps estimat per la tasca: 3h
  - Temps real: 1h
- US-058: Es mostren missatges de confirmació quan s'acaben els processos principals
  - Criteri d'acceptació:
    - Cada cop que acaba un procés important es mostra una finestra modal amb un missatge de notificació per tal que l'usuari pugui rebre feedback.
    - Aquests processos son: Crear i esborrar un projecte, crear i esborrar un requisit de negoci i fer anar el procés principal.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-059: La finestra de que mostra els atributs dels nodes desapareix al fer *drag*
  - Criteri d'acceptació:
    - Quan l'usuari fa clic sobre un node i el mou (*drag*) la finestra que mostra els atributs de la classe desapareix.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 2h

- US-060: Millora del *Look and Feel* del sistema
  - Criteri d'acceptació:
    - El sistema utilitza Bootstrap[30] i llibreries que defineixen diferents classes CSS[31, 32].
    - Els processos principals tenen una barra de progrés.
    - Tot el sistema té un aspecte agradable, utilitzant recursos CSS.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 20h
  - Temps real: 22h

### **User Stories Modificades**

- US-055: Llegenda interactiva per al visor de resultats
  - Motivació: No cal tant nivell de detall en els tipus de node dels grafs multidimensionals del visor de resultats.
  - Criteri d'acceptació:
    - A la part inferior del visor de resultats hi ha una llegenda de colors que marca de quin color és cada tipus de node: Conceptes i mesures.
    - Quan l'usuari fa clic sobre un dels colors es desplega una paleta de colors i l'usuari pot triar el color que desitgi.
    - Automàticament tots els nodes d'aquest tipus canvien el seu color al nou color triat.
  - Resultat: Acceptat

### **User Stories Posposades**

- US-028: Mostrar/amagar suggeriments de rol pels conceptes rellevants
  - Motivació: Per a dur a terme aquesta funcionalitat necessitem que el mòdul AMDO es trobi en un estat més avançat del desenvolupament, i donat que el projecte està a punt d'acabar, posposarem aquesta tasca fins que es pugui dur a terme (Secció 11).
- US-029: Explorar els conceptes de l'ontologia amb una representació tipus arbre
  - Motivació: Tal i com tenim la interfície gràfica dissenyada, no veiem clar quina seria la millor forma d'integrar aquesta funcionalitat, de forma que hem decidit posposar-la temporalment i tornar a treballar-ho en un futur (Secció 11).

## **7.12 Sprint 12**

Data de la reunió: 17/06/2014

**Resum de l'sprint:** Aquest és l'últim *sprint* del projecte. Amb aquesta reunió acabem la Fase 2 del projecte. Hem tractat i finalitzar les User Stories que ens quedaven pendents al Product Backlog.

Ara només resta acabar de preparar la documentació i el producte final, així com dur a terme el dia de test (Secció 9). Podem veure un resum del software final (arquitectura, tecnologies i mapa navegacional) i un exemple d'ús d'aquest a la Secció 8.

#### **User Stories tractades:**

- US-033: Exportar el procés ETL com un fitxer Kettle
  - Criteri d'acceptació:
    - El visor de resultats té un botó que permet a l'usuari descarregar el procés ETL relacionat amb la representació multidimensional que s'està visualitzant com un fitxer Kettle d'extensió .ktl
    - El fitxer es pot obrir amb el programa Kettle i representa amb exactitud el procés ETL que mostra el sistema.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 2h
  - Temps real: 3h
- US-034: Exportar l'esquema multidimensional com un conjunt de sentències DDL SQL
  - Criteri d'acceptació:
    - El visor de resultats té un botó que permet a l'usuari descarregar un conjunt de sentències DDL SQL.
    - Si s'executen aquestes sentències amb qualsevol intèrpret d'SQL s'obtindrà una reproducció de la interpretació multidimensional que s'està visualitzant al visor de resultats.
    - Les sentències es descarreguen en un arxiu amb extensió .txt.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h
- US-061: Canvi dinàmic de l'estat dels botons d'accions
  - Criteri d'acceptació:
    - Els botons d'accions del sistema canvien dinàmicament segons si es poden utilitzar o no:
      - Si no s'ha triat cap element de les llistes de les finestres d'obrir projecte, requisit o fitxer de resultats, el botó d'obrir romà desactivat.
      - Si no hi ha cap projecte al sistema, els botons d'obrir projecte i esborrar projecte de la pantalla principal romanen desactivats.
      - Si no hi ha requisits definits al projecte actual, els botons de Editar Requisit i Minar de la pantalla de gestió de projectes romanen desactivats.
      - Si no hi ha fitxers de resultats definits al projecte actual, el botó de obrir resultats romà desactivat.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h

- US-062: Les accions del MineDigger es mostren en un pane lateral desplegable
  - Criteri d'acceptació:
    - Al MineDigger es mostra un pane lateral parcialment amagat que es desplega quan el ratolí hi passa per sobre i es torna a plegar quan el ratolí surt de l'àrea d'aquest.
    - El pane conté totes les accions bàsiques del MineDigger: Guardar el graf com a imatge (US-056), comprovar la correctesa del requisit (US-020), amagar/mostrar nodes no rellevants (US-016/US-017), fixar/no fixar el graf (US-057), eliminar totes les modificacions fetes per l'usuari (US-019), tancar el MineDigger (US-026) i guardar el requisit (US-021).
  - Resultat: Acceptat
  - Temps estimat per la tasca: 6h
  - Temps real: 6h
  
- US-063: Esborrar fitxers de resultats
  - Criteri d'acceptació:
    - L'usuari pot triar un fitxer de resultats de la llista de fitxers de resultats generats del seu projecte.
    - Un cop triat, pot pitjar el botó *Delete* per a esborrar aquest requisit de la base de dades. Utilitzarem un *servlet* i una petició POST.
  - Resultat: Acceptat
  - Temps estimat per la tasca: 1h
  - Temps real: 1h

## 7.13 Product Backlog Final

Identificació	Nom	Estatus	Hores Estimades	Hores Totals
US-001	Adaptar l'estructura del graf per tal de poder visualitzar els components de l'ontologia.	OK	2	1
US-002	Crear un projecte nou	OK	1	4
US-003	Obrir un projecte existent	OK	1	1
US-004	Esborrar un projecte existent	OK	1	1
US-005	Canviar la Ontologia del projecte	OK	1	0,5
US-006	Canviar els <i>Source Mappings</i> del projecte	OK	1	0,5
US-007	Obrir MineDigger	OK	1	1
US-008	Fer funcionar el procés principal (GEM + ORE + COAL)	OK	10	15
US-009	Carrega dels resultats del procés principal	OK	1	0,5
US-010	Carregar Ontologies al MineDigger	OK	5	10
US-011	Manipular el Graf	OK	2	7
US-012	Marcar un concepte com a rellevant.	OK	5	10
US-013	Marcar un concepte com a no rellevant	OK	0,5	0,5
US-014	Definir el rol multidimensional d'un concepte	OK	7	10
US-015	Mostrar el rol definit pels conceptes rellevants	OK	2	3
US-016	Amagar els conceptes no-rellevants i no-relacionats	OK	2	2
US-017	Mostrar tots els conceptes	OK	1	1
US-018	Ressaltar una paraula clau del MineDigger	POSPOSAT	-	-
US-019	Esborrar totes les modificacions fetes al graf	OK	2	2
US-020	Comprovar la correctesa del requisit de negoci	OK	1	1
US-021	Generar un requisit de negoci a partir del graf	OK	4	3
US-022	Mostrar requisits prèviament generats.	OK	1	1
US-023	Carregar requisits prèviament generats.	OK	1	0,5
US-024	Esborrar Requisits prèviament generats.	OK	1	0,5
US-025	Canviar la Paleta de Color dels nodes	AMPLIAT - OK	-	-
US-026	Tancar l'editor	OK	0,5	0,5



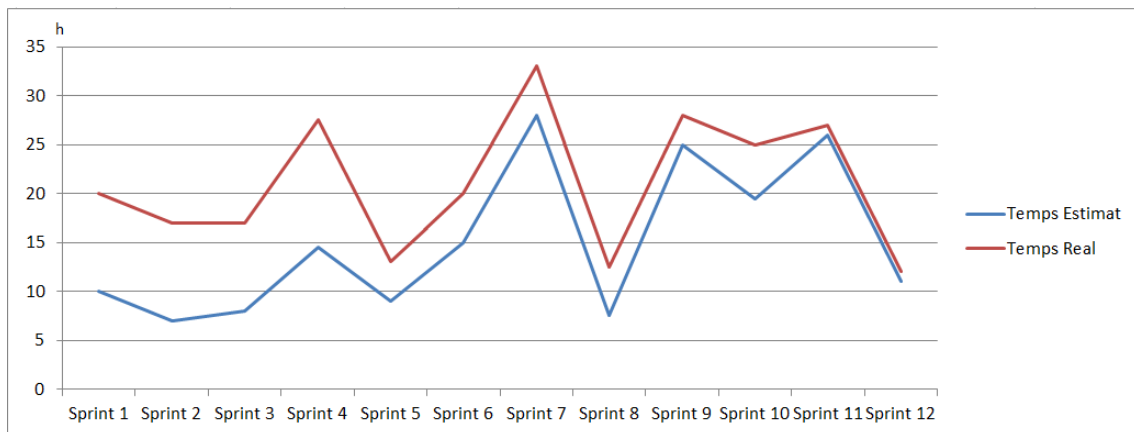
Identificació	Nom	Estatus	Hores Estimades	Hores Totals
US-027	Afegir accions sobre els nodes	AMPLIAT - OK	-	-
US-028	Mostrar/amagar suggeriments de rol pels conceptes rellevants	POSPOSAT	-	-
US-029	Explorar els conceptes de l'Ontologia amb una representació de tipus arbre	POSPOSAT	-	-
US-030	Carregar resultats	OK	4	5
US-031	Canviar entre els diferents diagrames al visor de resultats	OK	1	1
US-032	Exportar el graf multidimensional com una imatge	OK	1	2
US-033	Exportar el procés ETL com un fitxer Kettle	OK	2	3
US-034	Exportar el esquema multidimensional com un conjunt de sentències DDL SQL	OK	1	1
US-035	Ressaltar una paraula clau de visor de resultats	POSPOSAT	-	-
US-036	Tancar la pantalla de resultats	OK	0,5	0,5
US-037	La tria de l'eina de visualització és adequada	OK	10	20
US-038	S'han de poder guardar els canvis i l'estat del graf en un document JSON	OK	3	5
US-039	La tria de l'eina per al servidor és adequada.	OK	5	7
US-040	La visualització de l'ontologia és adequada i clara per a l'usuari	OK	1	1
US-041	Podem carregar qualsevol ontologia que tingui format .owl directament a la nostra eina.	OK	2	10
US-042	Disposem d'una base de dades interna NoSQL capaç de manegar les dades del sistema	OK	6	7
US-043	La base de dades ha de ser accessible des del Servidor	OK	3	3
US-044	La base de dades ha de ser accessible des del Client	OK	3	4
US-045	Cada concepte té associat una etiqueta que en mostra el nom.	OK	3	2
US-046	Apareixen requadres contextuais sobre els nodes al passar el ratolí per sobre.	OK	3	13
US-047	Marcar els descriptors rellevants d'un nivell	OK	15	20
US-048	Definir restriccions sobre els descriptors d'un nivell	OK	10	10
US-049	Definir diferents focus d'anàlisi sobre un fet	OK	10	13
US-050	Mostrar una llista d'atributs en una <i>box</i> quan aquests son massa llargs	OK	2	1
US-051	Mostrar una llegenda	AMPLIAT - OK	-	-
US-052	Es mostra de forma visual la jerarquia de classes al graf	OK	1	3
US-053	Exportar els fitxers ETL com imatges	OK	0,5	0,5
US-054	Llegenda interactiva per al MineDigger	OK	5	8

Identificació	Nom	Estatus	Hores Estimades	Hores Totals
US-055	Llegenda interactiva per al visor de resultats	OK	1	1
US-056	Exportar els requisits com imatges	OK	0,5	0,5
US-057	El graf interactiu pot anclar els nodes o be simular forces	OK	3	1
US-058	Es mostren missatges de confirmació quan s'acaben els processos principals	OK	1	1
US-059	La finestra de que mostra els atributs dels nodes desapareix al fer <i>drag</i>	OK	1	2
US-060	Millora del <i>Look and Feel</i> del sistema	OK	20	22
US-061	Canvi dinàmic de l'estat dels botons d'accions	OK	1	1
US-062	Les accions del MineDigger es mostren en un pane lateral desplegable	OK	6	6
US-063	Esborrar requisits i fitxers de resultats	OK	1	1
<b>Totals</b>			<b>180.5</b>	<b>252</b>

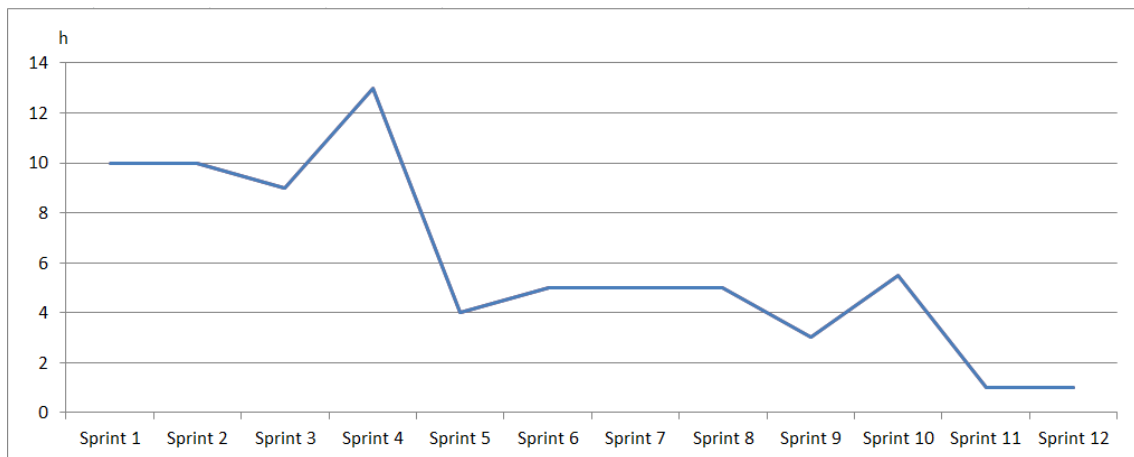
## 7.14 Gràfiques

A continuació les següents gràfiques mostren les relacions entre el treball estimat i el treball real. Podem veure que hi ha hagut *sprint* de més intensitat de treball (períodes festius i amb més disponibilitat per part dels recursos).

També podem veure que a mesura que el projecte ha anat avançant, les nostres prediccions s'han anat ajustant millor a la realitat. Amb l'ús de la metodologia i les tecnologies hem sigut capaços de poder predir amb major fiabilitat i, per tant, planificar millor els *sprints*, preveure la càrrega de treball que podíem suportar i prioritzar.



Gràfica 1: Comparativa entre el temps real de treball i les estimacions



Gràfica 2: Diferència entre el temps real de treball i les estimacions

## 8. Planificació Final

Hem pogut veure a la secció 3 tota la planificació inicial, i durant les seccions 5 i 6 hem vist com s'ha desenvolupat el projecte en les seves iteracions més importants. A continuació mostrem com ha restat la planificació després de tots els canvis que ha patit durant el projecte. Parlarem de les desviacions sofertes i com les hem assaltat, i finalment farem una revisió del pressupost del projecte final.

### 8.1 Desviacions

Les Fases 0 i 1 i s'han dut a terme sense patir cap desviació. La planificació era prou acurada i la naturalesa de les tasques no comportava massa risc. Hem seguit la planificació amb cura i les hem pogut realitzar amb èxit.

La Fase 2 es la que engloba la implementació, i en aquest punt si hem patit desviacions, cosa que per una banda és normal al utilitzar la metodologia *Agile*: El client decideix fer canvis, proposa noves funcionalitats, modifica alguns requisits en base als prototips que ha testat...

Durant tota la secció 7 queda detallat quines tasques s'han dut a terme en quin moment, i quan ha aparegut nova feina o s'ha cancel·lat algun requisit. Hem dut un control estricte de les hores invertides (que es pot consultar al Product Backlog final, Secció 7.13) i es pot veure als gràfics de la secció 7.14 com les nostres planificacions resultaven més acurades a cada *sprint*.

Cal remarcar com a desviació important que a meitat de la Fase 2 l'estudiant va patir una malaltia que el va impossibilitar durant gairebé dues setmanes fins al punt de requerir hospitalització. Els dies següents es va necessitar un treball de planificació important, i un estudi exhaustiu de les User Stories restants i del Product Backlog. Finalment com es pot veure a la secció 7, es va trobar la forma de repartir les Stories durant els *sprint* restants i finalitzar el projecte.

El contacte continu amb el client, el control estricte del product backlog i les tècniques descrites a la secció 3.4 son les eines que ens han permès manegar aquestes desviacions i assolir la implementació amb èxit. Durant el transcurs d'aquesta segona iteració el Diagrama de Gantt inicial ha patit modificacions (degut a aquestes desviacions).

A continuació mostrem l'aspecte del diagrama de Gantt al finalitzar el projecte. Com podem comprovar l'aspecte de la Fase 2 ha canviat completament. Hem reduït la granularitat de les tasques, ja que el nombre de tasques ha crescut en gran mesura i resulta més intuïtiu d'aquesta manera (i el control detallat de les subtasques tasques ja queda implícit a la Secció 7). També cal remarcar que la majoria de tasques de la Fase 2 se solapen les unes amb les altres, ja que al ser un procés iteratiu i basat en prototipatge ràpid, hem dut a terme moltes tasques en paral·lel, i alhora en paral·lel amb l'aprenentatge.

La Fase 3 ha patit unes poques desviacions a causa de la incorporació del dia de test i al conèixer la data final de la lectura del projecte. També cal remarcar que la generació de la documentació final va començar en paral·lel amb el final de la Fase 2, de forma que aquesta iteració es solapa amb els últims dies de la segona.

## 8.2 Diagrama de Gantt Final

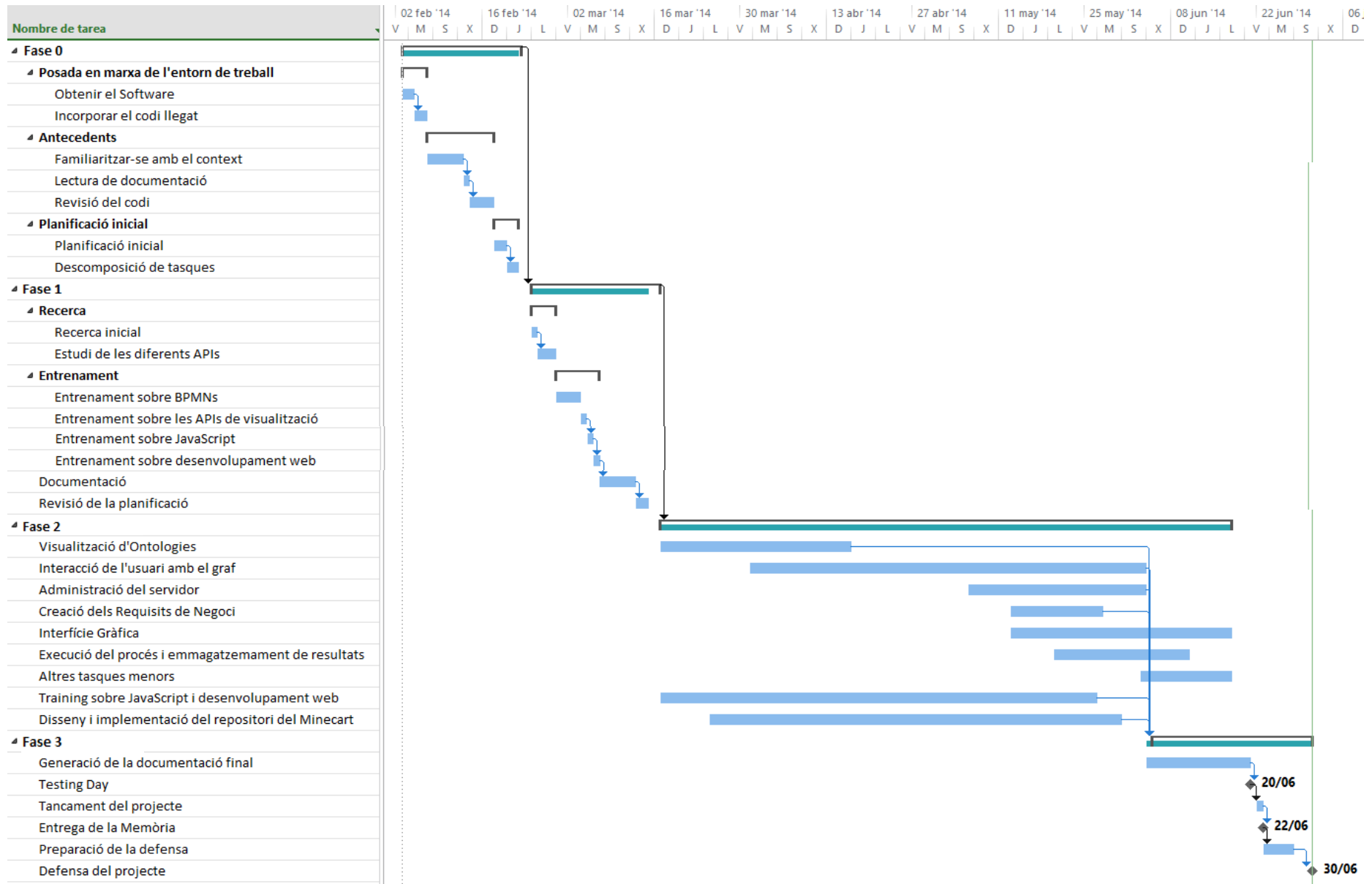


Figura 29: Diagrama de Gantt final del projecte [Font: Propi]

## 8.3 Pressupost Final

La següent taula mostra el pressupost final del projecte. Donat que no hem necessitat cap software nou ni cap llibreria de pagament, i la quantitat d'hores treballades no supera les planificades (Secció 7.13), concloem que no ens cal fer cap modificació amb el contracte inicial, i el pressupost final resta de la següent manera:

Concepte	Cost Estimat
Recursos Humans	4,400.00€
Hardware	143.52€
Software	86.02€
Despeses Generals	1.095 €
Total	5,725.54€

Taula 6: Pressupost Final

## 9. MineDigger i Minecart *in a Nutshell*

En aquesta secció tractarem de resumir els resultats del projecte. Mostrarem uns esquemes de l'arquitectura final, llistarem tot el conjunt de tecnologies que s'han utilitzat i mostrarem el mapa navegacional final de la interfície gràfica.

Finalment amb un conjunt d'imatges veurem un exemple del funcionament del software pas a pas.

### 9.1 Arquitectura

Aquest primer esquema mostra l'arquitectura en relació a la base de dades interna. Com podem veure, el nostre sistema utilitza un conjunt molt heterogeni de dades. En un primer nivell tenim fitxers de tot tipus: JSON, XML, javaObjects...

Aquests fitxers son parsejats a un format comú (o binary data) i emmagatzemats dins la base de dades interna, que s'estructura en col·leccions i dins de cada col·lecció, en documents:

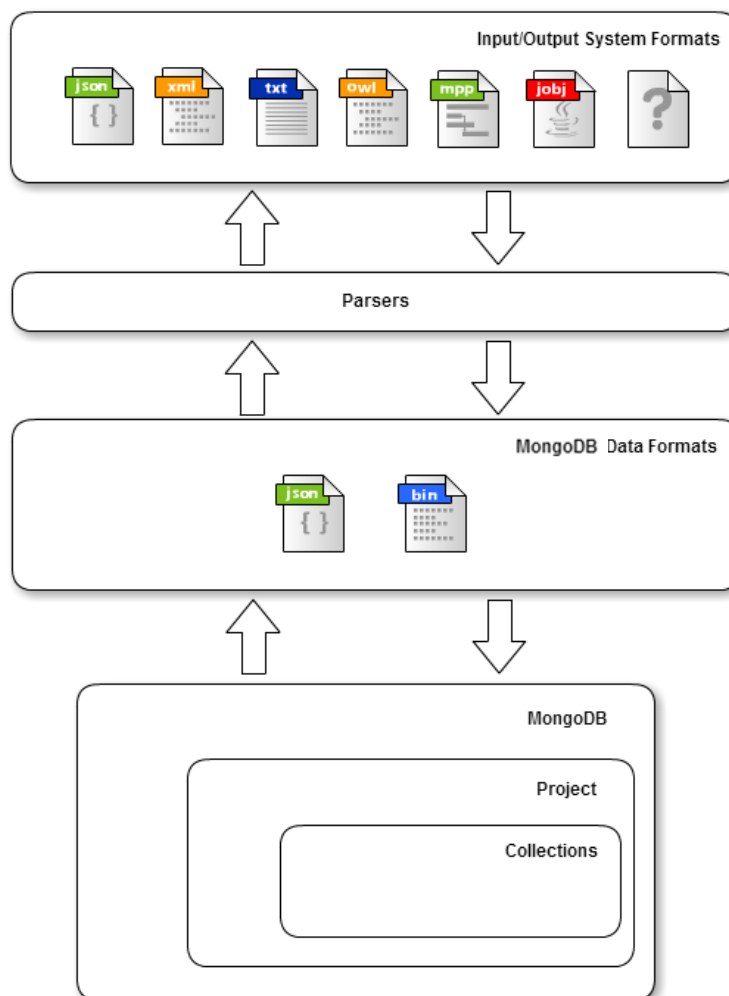


Figura 30: Arquitectura de la base de dades en relació amb els fitxers [Font: Propi]

Seguidament mostrem com s'estructura la base de dades internament a nivell de col·leccions. Podem veure que hi ha una col·lecció global de metadades, i després per a cada projecte trobem una col·lecció diferent segons el tipus de dades que s'emmagatzemen. Això ens és especialment útil quan hem de tractar amb les comunicacions i els servlets: simplifiquen les tasques i no perjudiquen el temps de resposta.

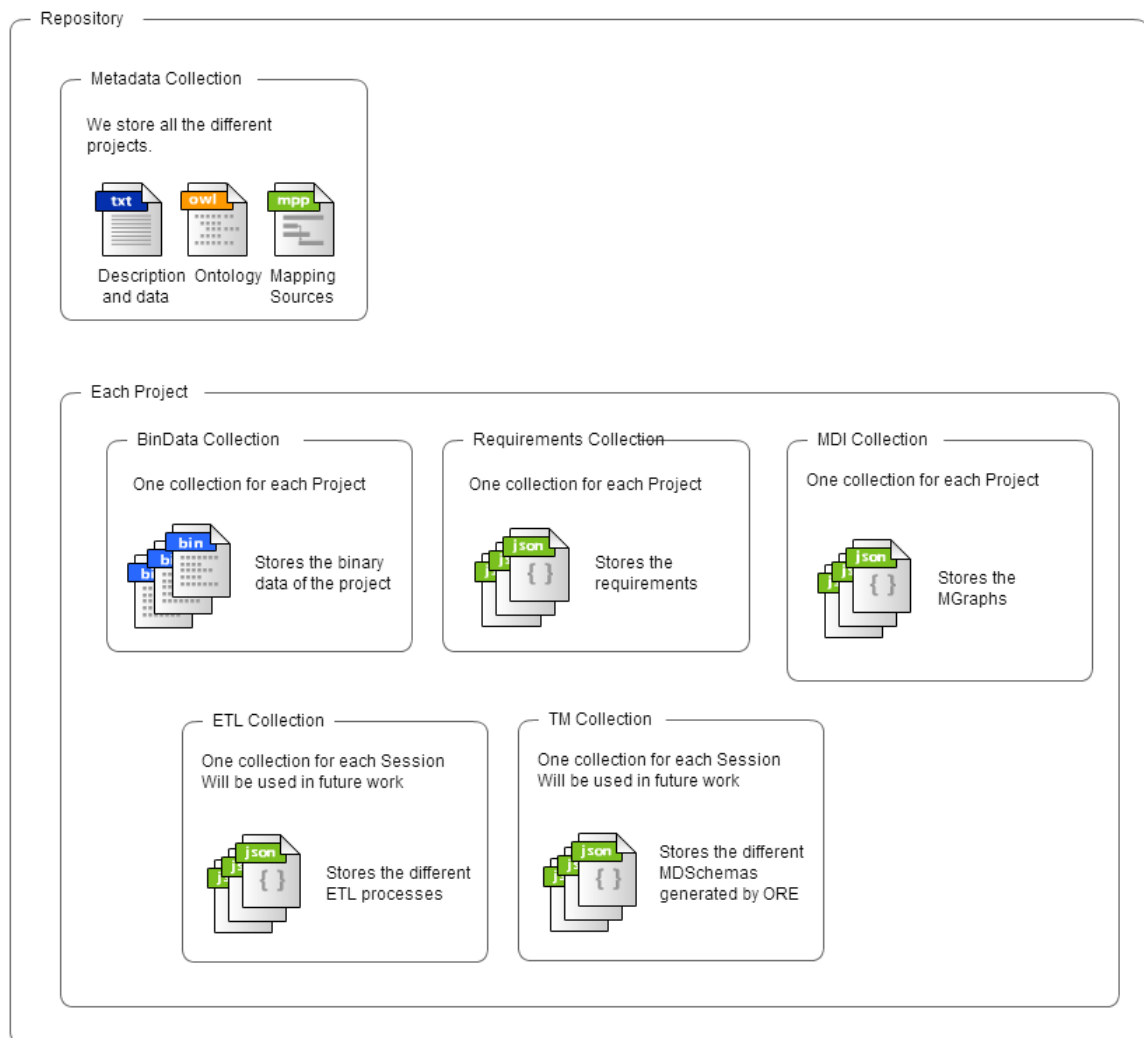


Figura 31: Arquitectura de la base de dades en relació amb les col·leccions [Font: Propi]

Finalment mostrem un esquema que engloba la totalitat del projecte. Podem veure la divisió client-servidor, la base de dades interna, tots els mòduls que componen el projecte GEM (fins i tot els que encara no estan totalment integrats com COAL), i els mòduls que hem desenvolupat durant aquest projecte (MineDigger i visor de resultats). També podem veure l'extensió d'aquest *Projecte Minecart*.



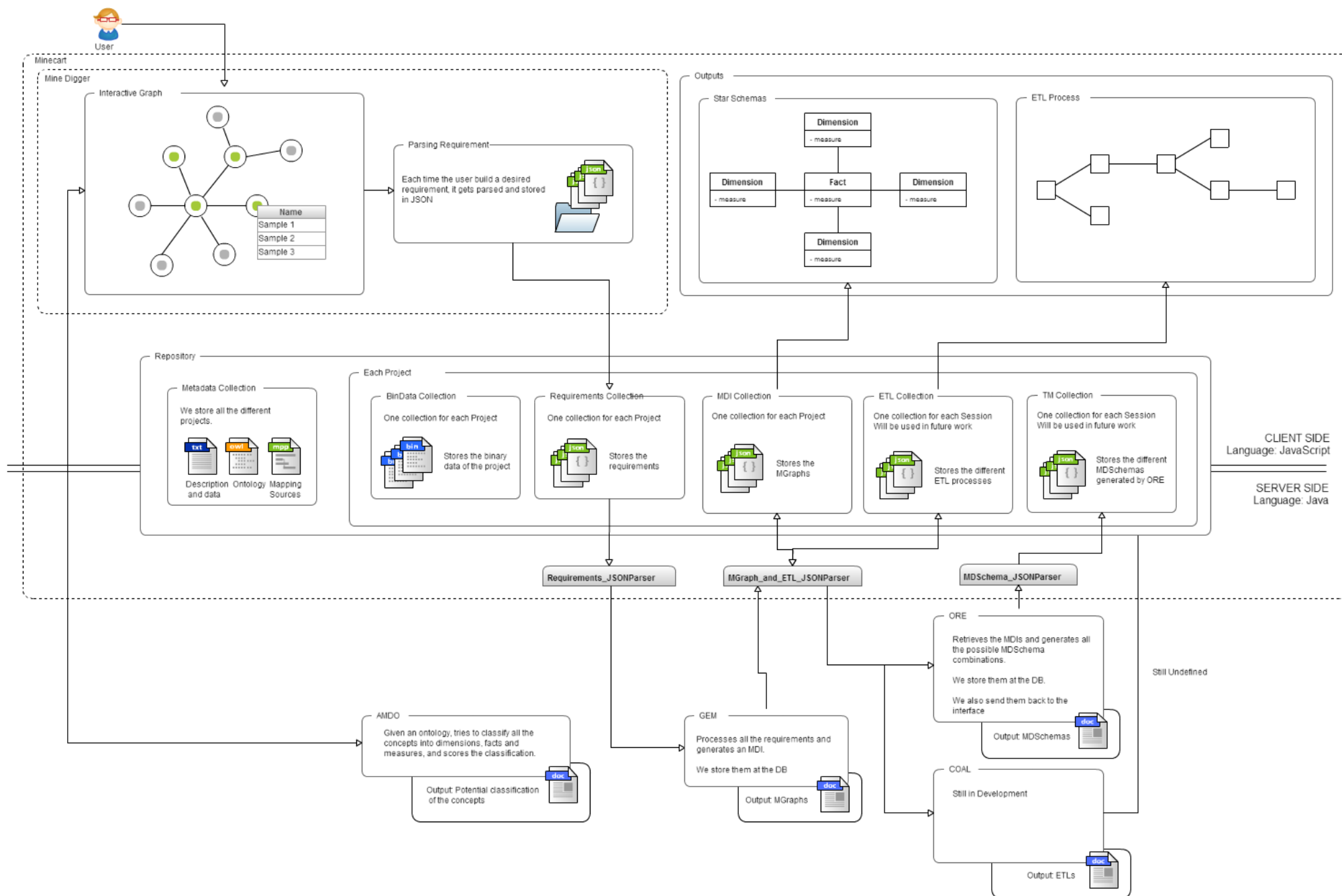


Figura 32: Arquitectura completa del sistema [Font: Propi]

## 9.2 Technologies

A continuació llistarem totes les tecnologies que hem utilitzat per a dur a terme el projecte (software, llibreries, eines..)

**Apache Tomcat 7.0.34:** Servidor capaç de fer anar servlets i JSP [29]

**Balsamiq Mockups:** Eina per a dissenyar mockups per a aplicacions web [32]

**Bootstrap:** Conjunt de fulles d'estil CSS que redefeixen algunes classes HTML [30]

**Bootswatch:** Fulla d'estil CSS que redefeix algunes classes de Bootstrap per a aconseguir una millora visual en alguns elements [31]

**Cacoo:** Eina online per al disseny de diagrames i treball col·laboratiu [33].

**D3.js:** Llibreria JavaScript emprada per a la visualització de tot tipus d'esquemes i grafs interactius [25]

**D3-Tip.js:** Llibreria JavaScript que afegeix finestres contextuais als elements d3.js [34]

**Eclipse IDE for Java EE Developers:** IDE emprat per a treballar amb Java[35]

**Jasny Bootstrap:** Llibreria que afegeix alguns elements extra a Bootstrap [32]

**Jena:** Llibreria encarregada de llegir i interpretar Ontologies [36]

**JIRA:** Eina online de gestió de projectes, integrada amb Subversion [19]

**JsColor:** Llibreria JavaScript emprada per a mostrar paletes de color interactives [37]

**Microsoft Office 365 Small Business Premium:** Conjunt d'eines per a elaborar documentació, gestionar projectes, i elaborar una gran varietat de diagrames [38]

**MongoDB:** Base de dades no relacional de clau-valor[23]

**Notepad++:** Editor de text orientat a la programació [39]

**Tortoise SVN:** Repositori Subversion[18]

**SubModal.js:** Llibreria JavaScript que ens permet crear finestres modals i definir respostes a les diferents opcions que s'hi puguin mostrar[40]

**Svg-Crowbar.js:** Llibreria JavaScript que transforma elements SVG a fitxers[41]

### 9.3 Mapa Navegacional Final

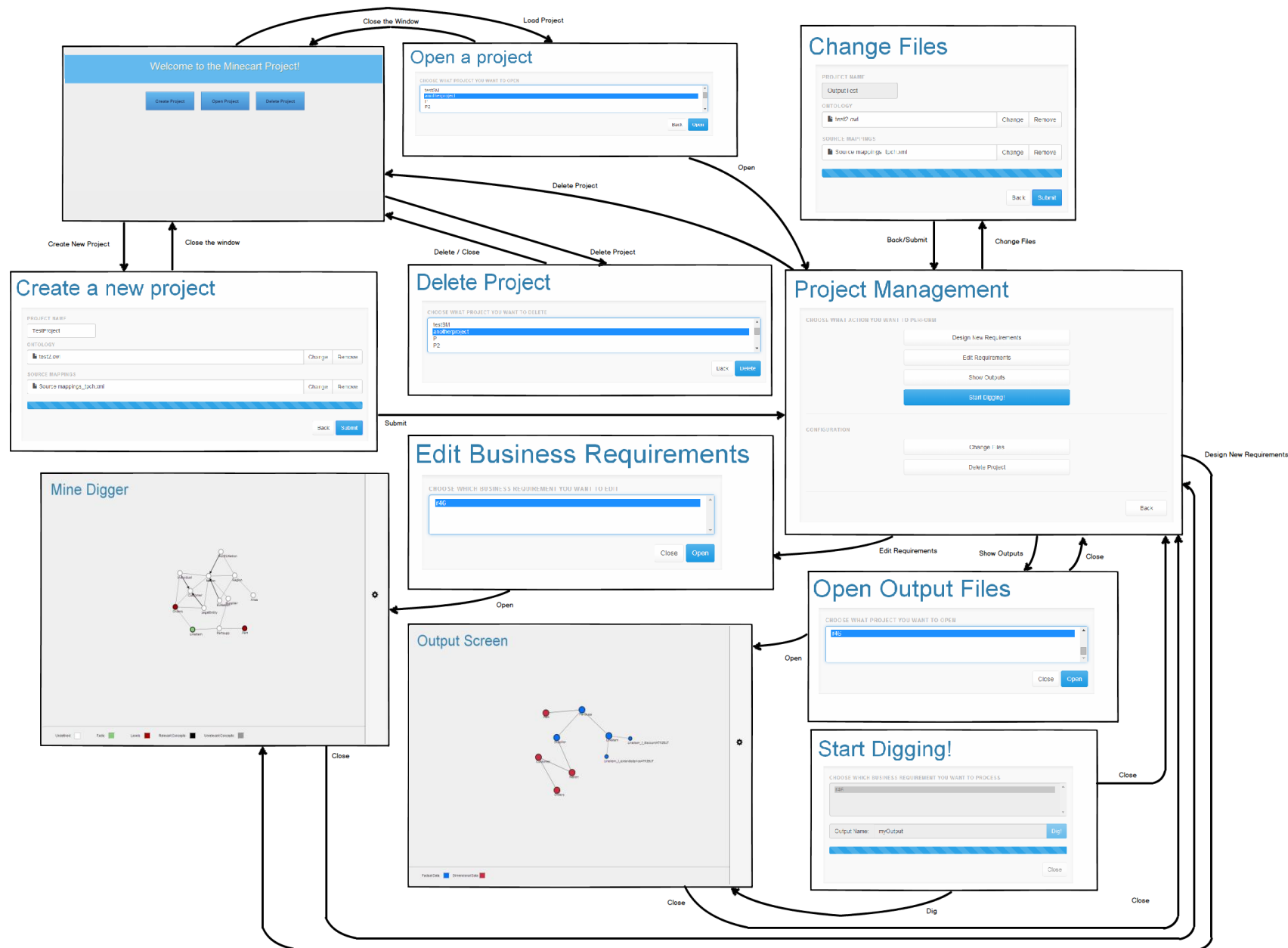


Figura 33: Mapa Navegacional final [Font: Propi]

## 9.4 Exemple de Flux de Treball

Per últim mostrarem un exemple d'ús del software. Crearem un projecte, dissenyarem un requisit i en generarem un fitxer de resultats.

1 – El primer pas serà crear el projecte. Obrim el software i pitgem el botó *Create Project*

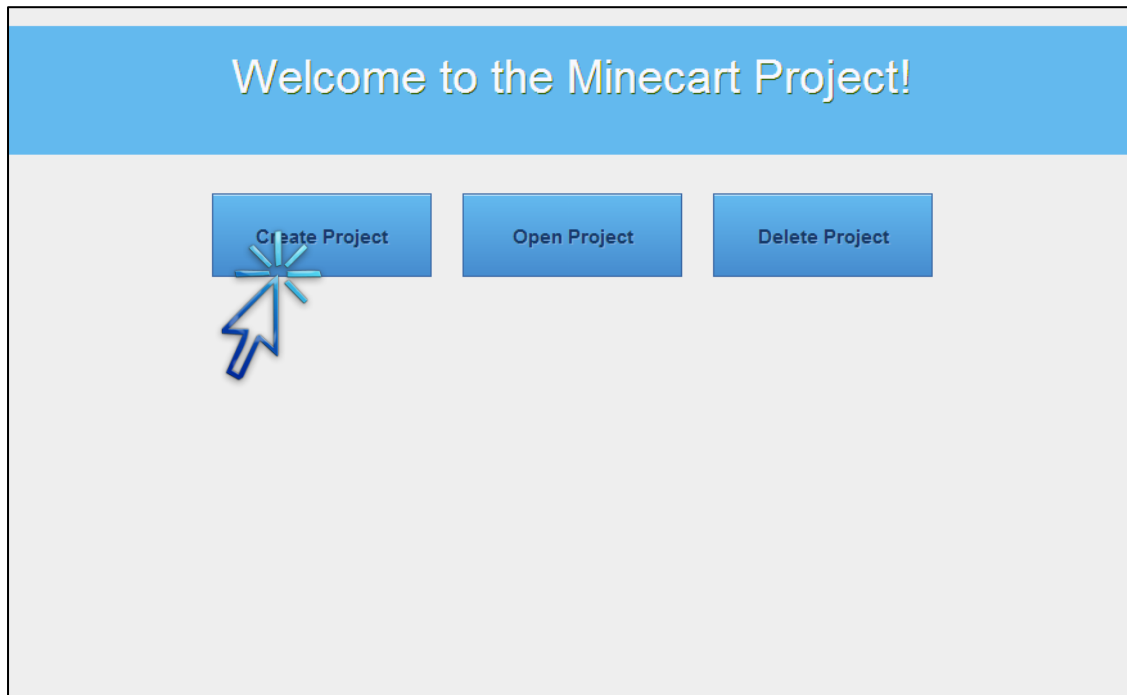


Figura 34: Finestra principal del projecte [Font: Propi]

2 – Seguidament cal emplenar les dades del projecte. Cal donar un nom i indicar quina és la Ontologia amb la que treballarem i el fitxer de mapeig de dades per al procés ETL. Un cop completat el formulari, farem clic al botó *Submit*.

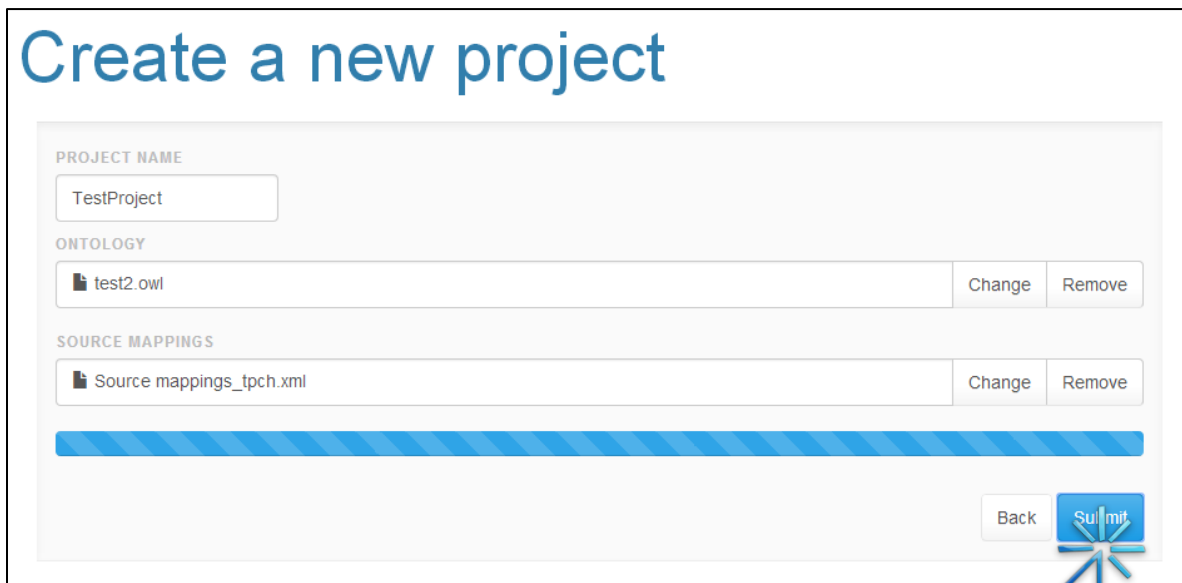


Figura 35: Finestra de creació de projecte [Font: Propi]

3 – Quan s’hagin carregat els fitxers al sistema, rebrem un missatge de confirmació, i seguidament passarem a la pantalla de gestió de projectes. Actualment no podem fer moltes accions, ja que no tenim cap requisit de negoci dissenyat.

Així, pitgem el botó *Design New Requirements*

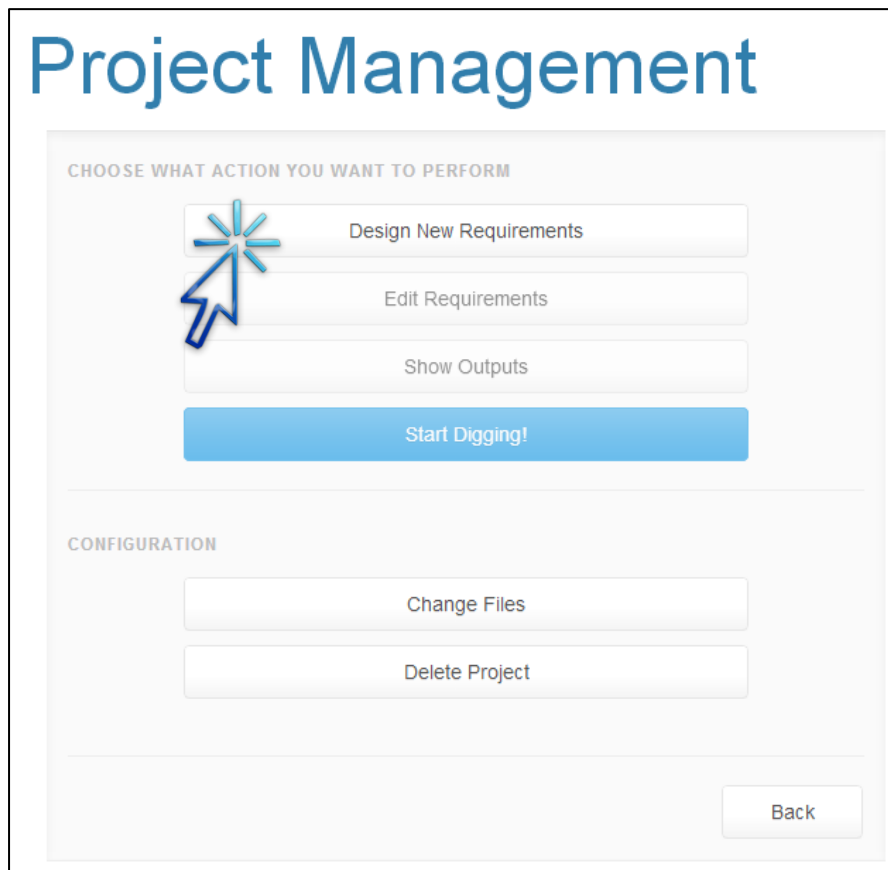


Figura 36: Finestra de gestió de projecte [Font: Propi]

4 – Ens trobem al MineDigger. Utilitzarem la finestra de configuració dels nodes per a definir les característiques del nostre requisit. Voldrem analitzar, per totes les ventes a partir del 01/01/2012, sota el punt de vista del nom del distribuïdor, la mitjana dels descomptes que s’han efectuat.

Per a definir el focus d’anàlisi cal clicar sobre el node *LineItem*, definir-lo com a fet o *Fact* i després clicar sobre *Add Focus*. Utilitzem el panell per a descriure la expressió, i no oblidem donar-li un al·lies.

Seguidament fem clic sobre el node *Supplier* i el definim com a nivell o *Level*. Marquem que ens interessa l’atribut *Supplier\_s\_name* ATRIBUT.

Finalment per a la restricció de data fem clic sobre el node *Orders* i el definim com a nivell o *Level*. Utilitzem el botó *Add Restriction* per a definir la restricció o *Slicer*.

Podem fins i tot canviar l’aparença del graf utilitzant la llegenda interactiva de la part inferior.

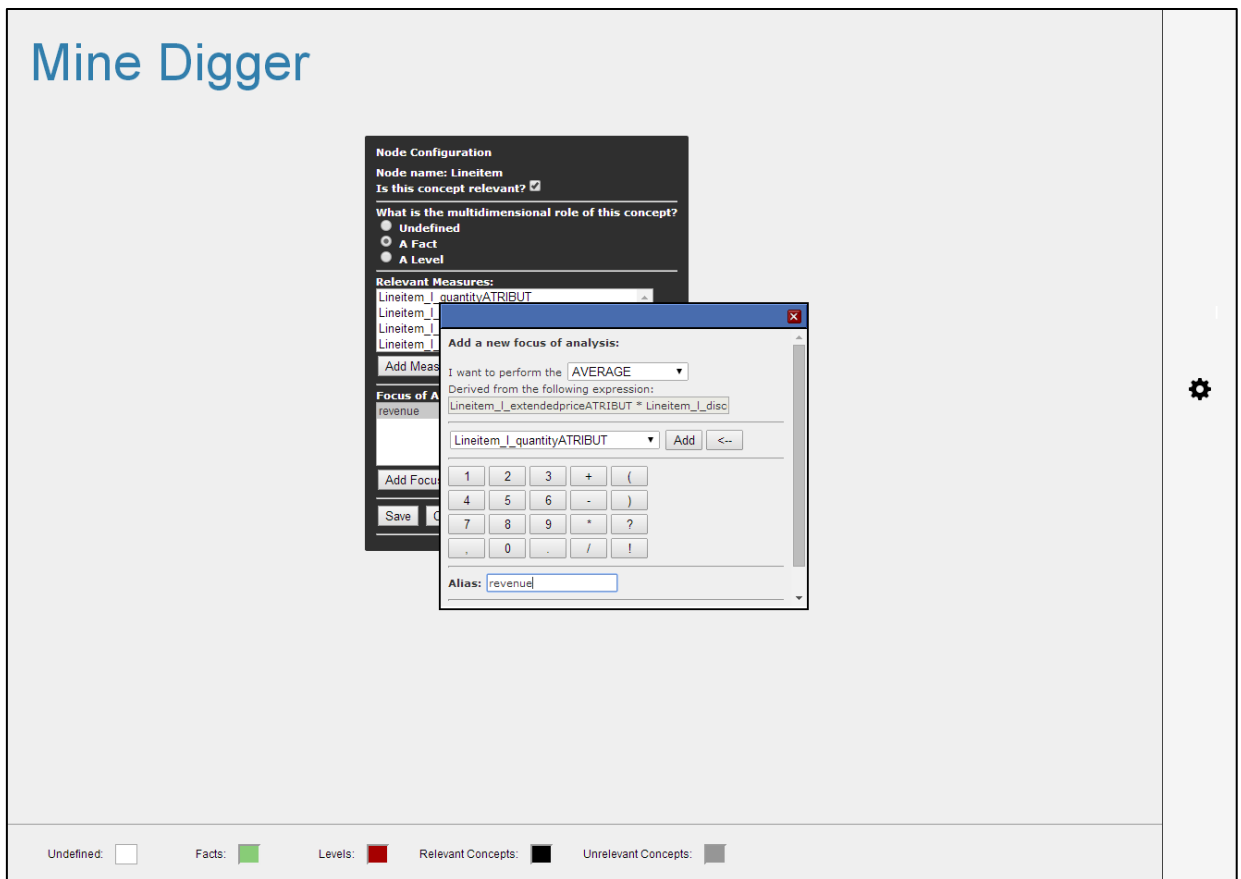


Figura 37: Disseny d'un focus d'anàlisi amb el MineDigger [Font: Propi]

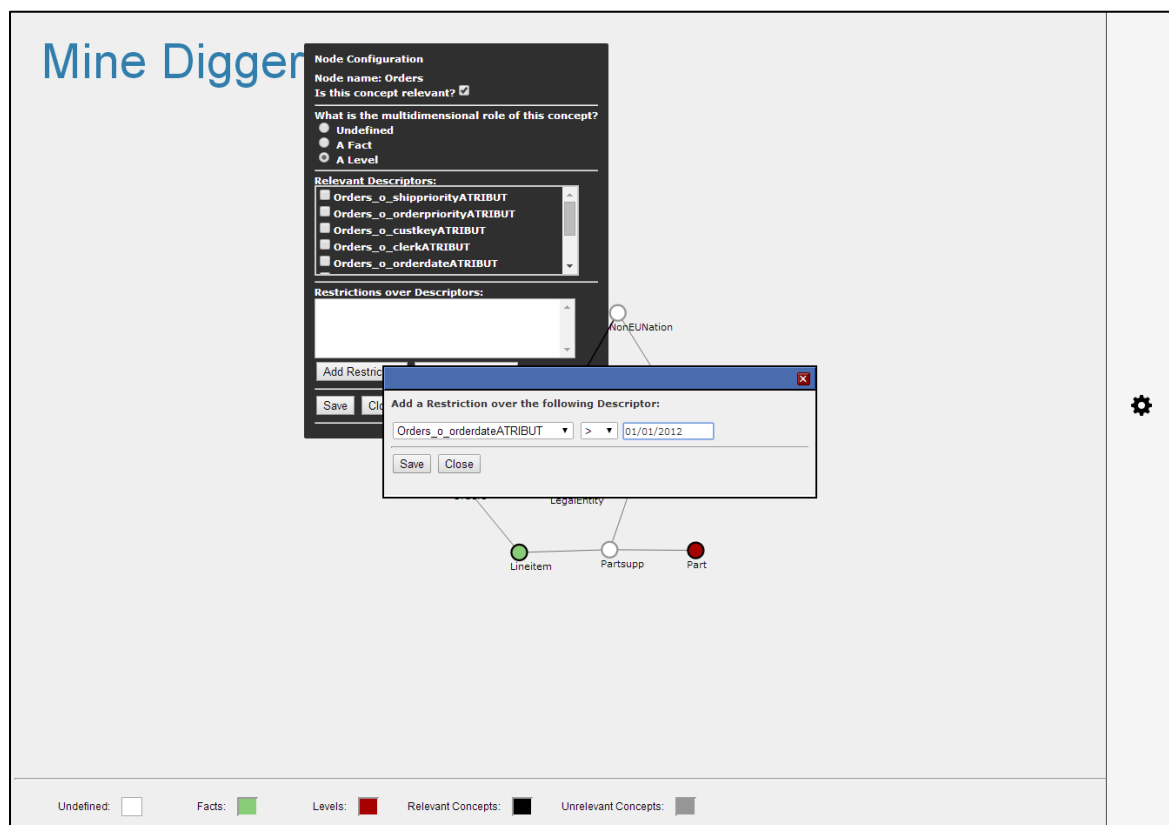


Figura 38: Disseny d'un slicer amb el MineDigger [Font: Propi]

5 – El requisit està acabat. Ara només cal obrir el pane lateral, donar-li un nom i pitjar *Save*. Al pane lateral trobem altres utilitats que ens permeten guardar una imatge del requisit, fixar-ne els nodes, mostrar només els conceptes rellevants, entre d'altres.

Un cop guardat el requisit se'ns mostrarà un missatge de confirmació i podrem seguir treballant sobre el graf. Nosaltres decidim tancar l'editor, ja que volem generar un fitxer de resultats.

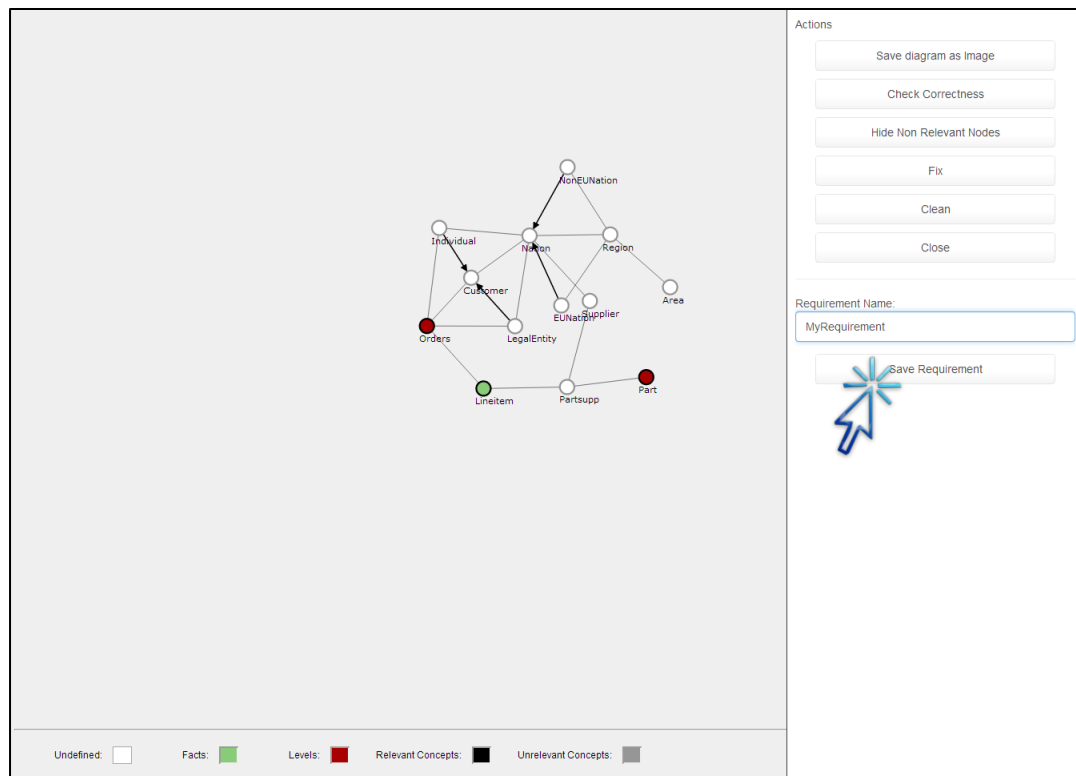


Figura 39: Creació del requisit de negoci amb el MineDigger [Font: Propi]

6 – A la finestra de gestió de projectes fem clic al botó *Dig*.

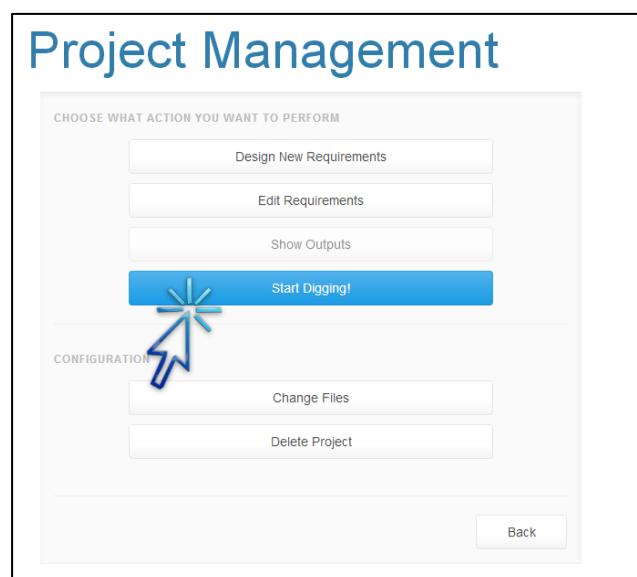


Figura 40: Finestra de gestió de projecte [Font: Propi]

7 – Triem quin requisit volem tractar i donem un nom al fitxer de sortida. Seguidament fem clic sobre el botó *Dig*.

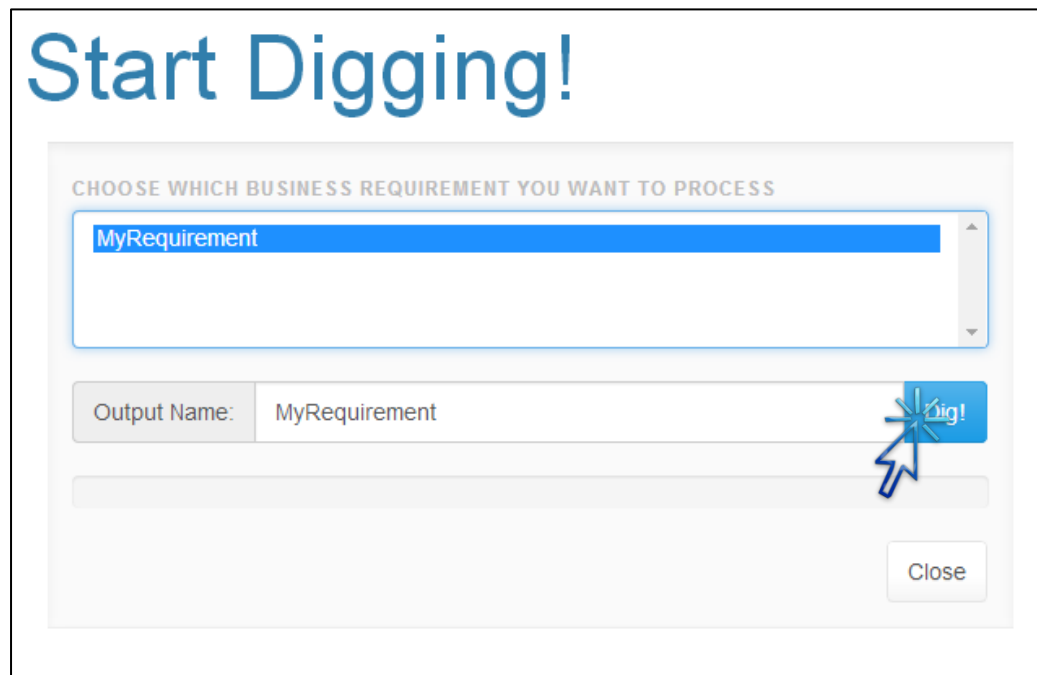


Figura 41: Finestra de càrrega de requisits al procés principal [Font: Propi]

8 – S'obrirà automàticament el visor de resultats. Aquí podem veure les interpretació multidimensional generada pel nostre software, així com els processos ETL associats. Podem navegar per aquests grafs a través del pane lateral, així com guardar-ne imatges i canviar-ne els colors.

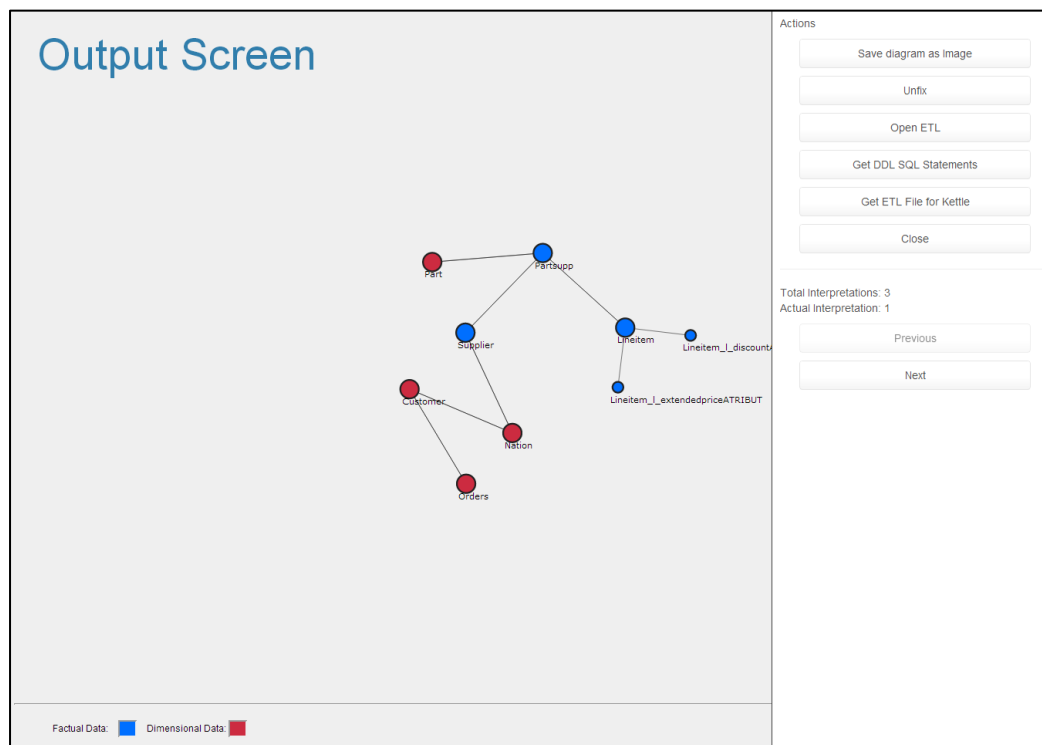


Figura 42: Visor de resultats – Interpretacions multidimensionals [Font: Propi]



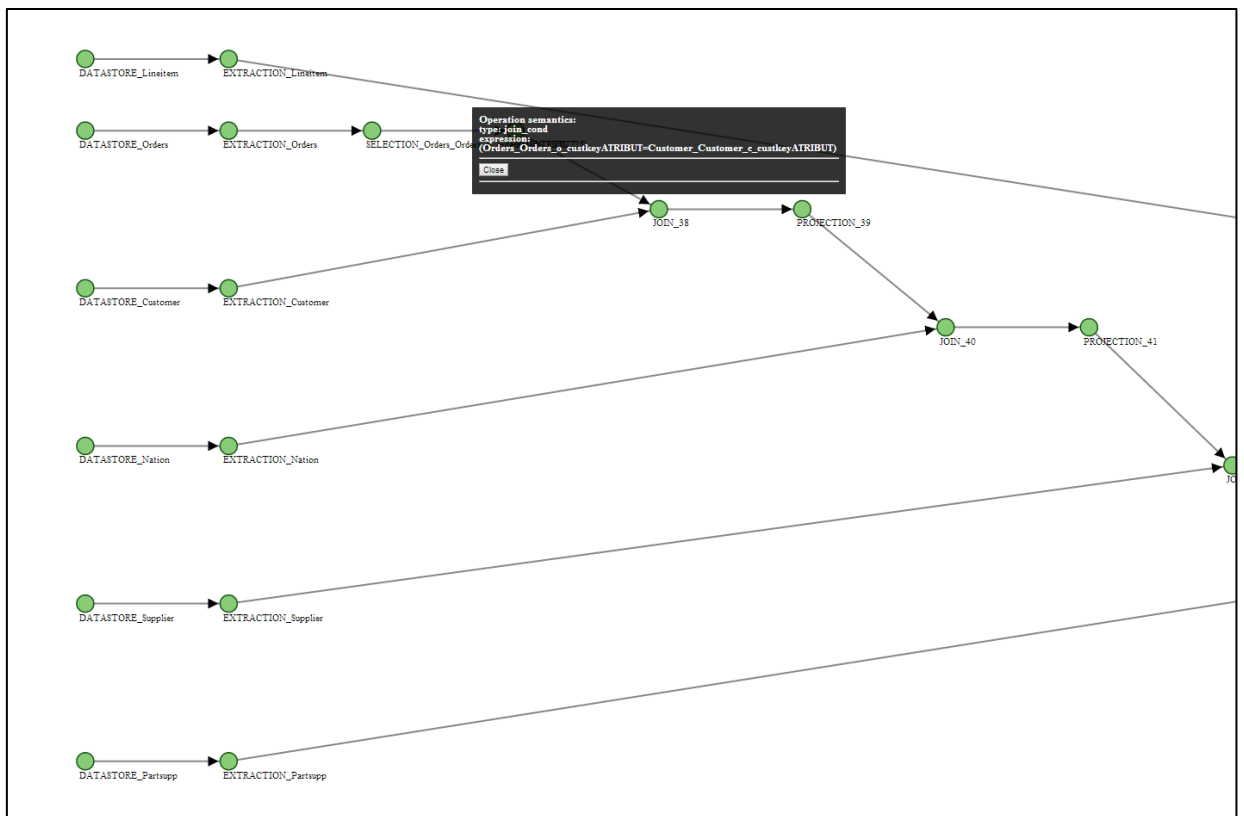


Figura 43: Visor de resultats – Processos ETL [Font: Propi]

# 10. Testing

Durant el transcurs del projecte hem dut a terme dos tipus de tests.

En primer lloc, durant la etapa **d'implementació** (Iteració 2) hem dut a terme reunions setmanals per a tancar cada *sprint*. En aquestes reunions es presentava una demo tècnica de les User Stories tractades aquella setmana. El client feia un test de les funcionalitats i donats uns criteris d'acceptació, validava o denegava la *user story*.

Durant aquest test podien aparèixer nous requisits, o podíem veure la necessitat de modificar-ne d'existents. És amb aquest contacte continu amb el client i amb el test que hem pogut definir el sistema a la mesura de les seves necessitats.

Adicionalment, durant el transcurs de la Iteració 3 hem realitzat un **dia de test**, el 20 de juny.

## 10.1 Dia de Test

### Participants

Els participants d'aquest dia de test son els següents:

- Alberto Abelló (professor, expert en BI)
- Vasileios Theodorou (estudiant doctorat, expert en BI)
- Jovan Varga (estudiant doctorat, expert en BI)
- Ruth Raventós (professora, experta en Software Engineering)
- Víctor Herrero (estudiant de màster, expert en data mining i business intelligence)
- Stephany Garcia (estudiant de màster, experta en data mining i business intelligence)

Tots ells disposen de diferents graus d'expertesa en relació al data warehousing i els esquemes multidimensionals, de forma que representen diversos perfils d'usuari. Hem cregut que resultaria útil veure el software des de diferents punts de vista, ja que un dels nostres objectius principals és que qualsevol usuari pugui utilitzar-lo.

### Entorn

L'entorn de test va ser a l'aula S205 de l'edifici Omega, al recinte de la UPC. Es requeria dur ordinador portàtil amb el navegador Google Chrome instal·lat.

S'ha utilitzat la Ontologia TCP *Benchmark* H (TPC-H) [43], un *benchmark* orientat a la presa de decisions. S'ha muntat el servidor del sistema i el servidor de MongoDB en un ordinador portàtil, utilitzant la Xarxa Sense Fils de la UPC com a xarxa local per a comunicar els ordinadors dels participants del test.

A cada participant se li ha registrat al sistema un projecte personal on poder dissenyar requisits i treballar.

## Desenvolupament del test

El test consta de tres parts:

Primera part: Els participants obren el seu projecte i comencen a familiaritzar-se amb el software. L'objectiu és veure com de senzill els resulta d'interpretar el context del projecte sobre el que treballaran (la Ontologia TPC-H) amb la seva representació gràfica.

Segona part: Es proposa un exercici per als participants: Cal dissenyar una requisit de negoci utilitzant el software que equivalgui a la següent frase:

*“Vull analitzar la **mitjana** dels beneficis de la meua promoció de **descomptes** Nadalencs, donats el **conjunt de productes demanats** abans de nadal l'any passat (25/12/2013), des del punt de vista del tipus de **part** del producte.*

Passat uns minuts s'ofereixen un conjunt de pistes que poden facilitar la feina als participants:

- Els beneficis es poden derivar de la següent expressió:  $AVERAGE(l\_extendedprice * l\_discount)$
- Una restricció pot ser  $l\_orderdate < 25/12/2013$
- Dins el node part ens pot interessar l'atribut p\_type.

L'aspecte que té la solució d'aquest exercici és la següent:

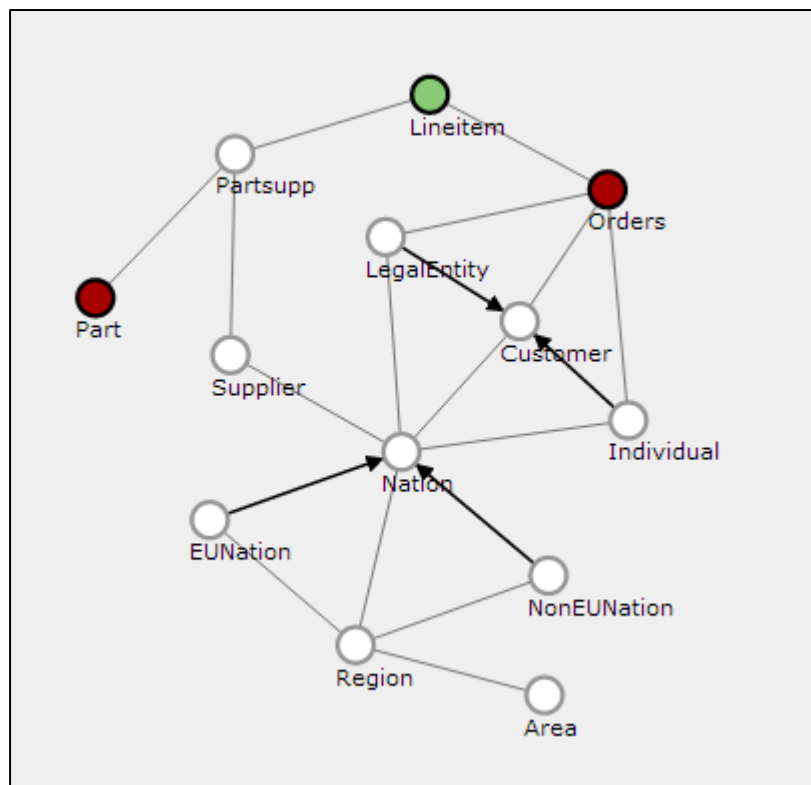


Figura 44: Requisit a dissenyar a la segona part del dia de test [Font: Propi]

Quan el requisit estigui construït, cal fer anar el sistema per a generar les propostes multidimensionals i els grafs ETL.

Tercera Part: Es deixa uns minuts als participants per tal de que experimentin amb el programa i provin de dissenyar els seus propis requisits.

### **Feedback**

Un cop acabada la sessió es dona als participants un qüestionari que poden respondre per transmetre les seves impressions. Les preguntes del qüestionari son les següents:

1. Indica el teu camp d'expertesa:
  - a. Bases de dades/data warehousing
  - b. Enginyeria de requisits
  - c. Modelat conceptual
  - d. *Business Intelligence*
2. Fins a quin punt ets familiar amb el camp del data warehousing? (0-4)
3. Fins a quin punt ets familiar amb el disseny de processos d'extracció, transformació i càrrega de dades (ETL)? (0-4)
4. Fins a quin punt ets familiar amb el disseny d'esquemes multidimensionals? (0-4)
5. Indica el nivell d'esforç requerit per tal de agafar familiaritat amb el domini (utilitzant la representació gràfica de la ontologia). (0-4)
6. Indica el nivell d'esforç requerit per tal de definir requisits textuais utilitzant la interfície gràfica del MineDigger. (0-4)
7. Indica el nivell d'esforç requerit per tal de poder explorar el domini i definir els teus propis requisits de negoci utilitzant el MineDigger. (0-4)
8. Fins a quin punt consideres l'esforç inicial per a utilitzar el MineDigger necessari? (0-4)
9. Fins a quin punt consideres que el MineDigger resulta intuïtiu per a usuaris no familiaritzats amb els Data Warehouses o les Bases de Dades? (0-4)
10. Indica l'esforç que comporta el procés sencer, des de definir els requisits fins a obtenir els dissenys finals. (0-4)
11. Fins a quin punt consideres que el sistema pot resultar útil per a casos reals on els projectes es troben en les seves primeres fases de disseny del data warehouse? (0-4)
12. Elabora, si cal, la resposta anterior.
13. Indica, si ho creus convenient, comentaris, impressions o suggerències per a millorar el sistema.

### **Resultats**

Un cop rebuts els qüestionaris, hem fet un petit anàlisi i per a extreure conclusions. Hem pogut veure que hi havia dos grups clarament diferenciats entre els participants: Aquells que coneixen el domini del sistema i aquells altres que no.

Les valoracions del nostre software han sigut molt positives per ambdós grups, però aquells que no coneixien el domini han tingut algunes dificultats extra els primers minuts.

Hem pogut comprovar que es tracta d'un tema d'intuitivitat i semàntica, i hem demanat consell i comentaris sobre possibles millores.

Alguns dels comentaris son els següents:

- “Seria positiu donar una descripció inicial de les semàntiques de la ontologia i dels conceptes bàsics del data *warehousing* (fets, nivells, mesures...)”
- “L'únic que em va costar de comprendre del software va ser el vocabulari emprat. Potser fer servir paraules menys tècniques com *slicer* i renombrar-los a frases més auto-descriptives com *Afegir una condició sobre l'atribut* o [...] d'aquesta manera serà més clar tant per a usuaris habituals de DW com per a usuaris sense coneixement al món del DW.”
- “Seria una bona idea introduir un vídeo amb un exemple d'ús del software, de forma que ràpidament l'usuari pugui veure com és el *workflow* bàsic de la aplicació”

També val a dir que els que coneixien el domini han pogut dissenyar els requisits de negoci de forma molt ràpida i es troben molt satisfets amb l'eina.

En qualsevol cas vam obtenir un feedback molt valuós que sens dubte serà d'utilitat per a millorar el programa, ja que hi els seguirem treballant un cop acabat el treball de final de grau, en projectes futurs (Secció 11).

## 11. Treball Futur

Un cop acabat el projecte, cal començar a pensar en les següents passes. Aquest projecte ens ha permès donar consistència i robustesa al projecte GEM, així com conciliar totes les seves parts i alhora ens ha obert les portes per a seguir treballant-hi i establir noves funcionalitats.

Com comentàvem a l'Estat de l'Art, alguns dels mòduls que executen els algorismes encara es troben en desenvolupament i el projecte es troba en constant evolució. L'AMDO és un mòdul funcional però cal pensar quina és la millor forma d'integrar-lo amb el sistema, ja que es va desenvolupar fa anys i hi ha hagut molts canvis a l'ecosistema. COAL encara es troba en les primeres passes del desenvolupament i ORE es troba en la seva etapa final.

A mesura que els responsables d'aquests mòduls vagin desenvolupant-los i el projecte avanci, caldrà ajustar i adaptar el projecte Minecart, com per exemple mostrar els suggeriments de l'AMDO d'una forma interactiva per a facilitar l'experiència de l'usuari, desplegar la memòria cau interna que utilitzarà aquest mòdul, entre d'altres.

Alhora permetrem que els usuaris puguin donar ús del nostre software. Això implicarà un seguit de tasques de manteniment: poden sorgir problemes, algunes funcionalitats es poden replantejar...

Alhora, com hem comentat a la secció 10, treballarem sobre les opinions donades pels participants del dia de test per poder millorar aquells detalls que ens van fer arribar durant les seves primeres impressions amb el software.

Finalment caldrà plantejar en un desplegament sobre el núvol, ja que com comentàvem a la Secció de sostenibilitat, suposarà una reducció de l'impacte ecològic.

Per a totes aquestes tasques l'estudiant continuarà treballant per al departament d'ESSI durant els mesos de juliol, agost i setembre sota un contracte de becari amb la UPC.

## 12. Conclusions

En aquest apartat compartiré les conclusions que crec que es poden extreure d'aquests cinc mesos de treball en el *Projecte Minecart*:

El primer que cal dir és que hem aconseguit el nostre objectiu: El projecte s'ha completat a temps, el client ha quedat satisfet i podem considerar que ha resultat un èxit. Tot i pensar alguns cops que ens veuríem forçats a treure funcionalitats o reduir el projecte degut a la càrrega de treball i als entrebancs amb els que ens hem trobat, finalment hem sortit endavant.

Hem hagut d'enfrontar-nos amb un projecte que ens ha fet treballar amb més de 4 anys de codi llegat, amb tecnologies desconegudes fins al moment i amb les quals teníem zero expertesa, amb una metodologia diferent a la coneguda, amb interacció constant amb el client, amb desviacions i contratemps continus...

En resum hem hagut de realitzar un projecte real, amb una aplicació real i en un ambient que bé podria ser el d'una empresa real. Ha sigut doncs aquest projecte un primer apropament al món laboral i a la recerca.

A nivell personal crec que ha estat una molt bona oportunitat per aprendre moltes coses útils, no només a nivell de llenguatges i tecnologies (que també), sinó a nivell de gestió de projectes. I sobretot l'experiència, que no és pas comparable amb la de qualsevol altre pràctica o projecte que hàgim fet en l'entorn universitari fins al moment.

Finalment comentar que començar aquest projecte ha obert alhora moltes portes i seguirem treballant en l'ecosistema GEM durant els propers mesos. Creiem que pot ser un sistema que pot resultar interessant a un públic molt ampli, i pot tenir molt de futur. Els participants del dia de test van quedar realment molt satisfets del nostre software, i vam poder veure amb els nostres propis ulls com podien realitzar, en pocs minuts, tasques que sense el software requereixen de personal expert i molt de temps i esforç.

Podem concloure, doncs, que el *Projecte Minecart* ha sigut tot un èxit.

## 13. Glossari d'Abreviatures

AMDO: Automatic Multidimensional Design from Ontologies

COAL: ALgorithm for COnsolidating ETL flows

CSS: Cascade StyleSheet

DB: DataBase

DW: Data Warehouse

ETL: Extraction, Transformation and Load

GEM: Generating ETL and Multidimensional designs

HTML: HyperText Markup Language

JS: JavaScript

MDI: MultiDimensional Interpretation

MGraph: Multidimensional Graph

NoSQL: Not Only Structured Query Language

ORE: Onthology-based data warehouse REquirement-driven evolution and Integration

OWL: Ontology Web Language

SM: Source Mappings

SVN: SubVersion

TM: Traceability Metadata



# 14. References

- [1] W. H. Inmon, What is a Data Warehouse? Prism Tech Topic, Vol. 1, No. 1, 1995
- [2] Petar Jovanovic, Oscar Romero, Alkis Simitsis, Alberto Abelló, Daria Mayorova: A Requirement-Driven Approach to the Design and Evolution of Data Warehouses. Elsevier 2014,
- [3] Data Warehouse Architecture. Digital image. Data Warehouse 4U. N.p., 2008. Web. 28 Feb. 2014. <<http://datawarehouse4u.info/>>.
- [4] Fact Constellation Schema. Digital image. Data Warehouse 4U. N.p., 2008. Web. 28 Feb. 2014. <<http://datawarehouse4u.info/>>.
- [5] Petar Jovanovic, Oscar Romero, Alkis Simitsis, Alberto Abelló: Integrating ETL Processes from Information Requirements. DaWaK 2012: 65-80
- [6] M. Golfarelli, D. Maio, S. Rizzi, The Dimensional Fact Model: a Conceptual Model for Data Warehouses, Int. J. Coop. Inf. Syst. 7 (2–3) (1998) 215–247.
- [7] C. Phipps, K.C. Davis, Automating Data Warehouse Conceptual Schema Design and Evaluation, Proc. of 4th Int. Workshop on Design and Management of Data Warehouses, volume 58, CEUR-WS.org, 2002, pp. 23–32.
- [8] Husemann, B., Lechtenborger, J., Vossen, G. Conceptual Data Warehouse Modeling. In: DMDW. 2000, p. 1:11.
- [9] Maz\_on, J.N., Trujillo, J.. An MDA Approach for the Development of Data Warehouses. DSS 2008;41-58.
- [10] Luja\_n-Mora, S., Vassiliadis, P., Trujillo, J.. Data Mapping Diagrams for Data Warehouse Design with UML. In: ER. 2004, p. 191-204.
- [11] Romero, O., Abelló, A., A Framework for Multidimensional Design of Data Warehouses from Ontologies. Data Knowl Eng 2010;69(11):1138-1157.
- [12] Romero, O., Simitsis, A., Abell\_o, A.. GEM: Requirement-driven Generation of ETL and Multidimensional Conceptual Designs. DaWaK. 2011, p. 80-95.
- [13] K Morton, M Balazinska, D Grossman, J Mackinlay, Support the Data Enthusiast: Challenges for Next-Generation Data-Analysis Systems, Proceedings of the VLDB Endowment 7 (6)
- [14] Trail: Creating a GUI With JFC/Swing (The Java™ Tutorials) <<http://docs.oracle.com/javase/tutorial/uiswing/>>
- [15] *Data Integration | Pentaho Community* <<http://community.pentaho.com/projects/data-integration/>>
- [16] Manifesto for Agile Software Development < <http://agilemanifesto.org/>>
- [17] Principles behind the Agile Manifesto <<http://agilemanifesto.org/principles.html>>
- [18] *TortoiseSVN – Home* <<http://tortoisesvn.net/>>
- [19] *Issue & Project Tracking Software | Atlassian* <<https://www.atlassian.com/software/jira>>

- [20] Iron triangle of Waterfall Software Development. Digital image. Agile Bench. August 2013 <<http://agilebench.com/>>.
- [21] The Era of Business Intelligence for the Masses has Dawned According to Info-Tech Research Group <<http://www.prweb.com/releases/2012/1/prweb9106634.htm>>
- [22] Technology Research | Gartner Inc <<http://www.gartner.com/technology/home.jsp>>
- [23] *MongoDB* <<http://www.mongodb.org/>>
- [24] Index of /download/protege/3.5 <<http://protege.stanford.edu/download/protege/3.5/>>
- [25] D3.js - Data-Driven Documents <<http://d3js.org/>>
- [26] Gallery · mbostock/d3 Wiki · GitHub <<https://github.com/mbostock/d3/wiki/Gallery>>
- [27] Play Framework - Build Modern & Scalable Web Apps with Java and Scala <<http://www.playframework.com/>>
- [28] node.js <<http://nodejs.org/>>
- [29] Apache Tomcat - <<http://tomcat.apache.org/>>
- [30] Bootstrap <<http://getbootstrap.com/>>
- [31] *Bootswatch: Free themes for Bootstrap* <<http://bootswatch.com/>>
- [32] *Jasny Bootstrap* <<http://jasny.github.io/bootstrap/>>
- [33] *Balsamiq Mockups – Balsamiq* <<http://balsamiq.com/products/mockups/>>
- [34] *Cacoo: Online Diagram Software for Flow Chart & UML and More* <<https://cacoo.com/>>
- [35] *Caged/d3-tip · GitHub* <<https://github.com/Caged/d3-tip>>
- [36] *Eclipse IDE for Java Developers | Eclipse Packages* <<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/keplersr2>>
- [37] *Apache Jena* <<https://jena.apache.org/>>
- [38] *JSColor – JavaScript / HTML Color Picker, Selector, Chooser* <<http://jscolor.com/>>
- [39] *Microsoft Office 365 para hogar o empresas – Office en linea* <<http://office.microsoft.com/es-es/products/?CTT=97>>
- [40] *Notepad++ Home* <<http://notepad-plus-plus.org/>>
- [41] *submodal - JavaScript / DHTML modal window solution - Google Project Hosting* <<https://code.google.com/p/submodal/>>
- [42] *SVG Crowbar* <<http://nytimes.github.io/svg-crowbar/>>
- [43] *TPC-H – Homepage* <<http://www.tpc.org/tpch/>>